

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Механіко-машинобудівний інститут
Кафедра технології машинобудування**

«На правах рукопису»
УДК 62-503.57

До захисту допущено:
Завідувач кафедри
_____ Юрій ПЕТРАКОВ
«15» травня 2020 р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-науковою програмою «Технології машинобудування»
зі спеціальності 131 «Прикладна механіка»
на тему: «Використання алгоритмів штучного інтелекту в
дослідженнях технологій машинобудування»**

Виконав (-ла):
студент (-ка) VI курсу, групи МТ-81мн
Артюшенко Віктор Владиславович _____

Керівник:
доцент кафедри, кандидат технічних наук, доцент,
Фролов Володимир Костянтинович _____

Рецензент:
професор кафедри, доктор технічних наук, професор,
Шевченко Олександр Віталійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студент (-ка) _____

Київ – 2020 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Механіко-машинобудівний інститут

Кафедра технології машинобудування

Рівень вищої освіти – другий (магістерський)

Спеціальність – 131 «Прикладна механіка»

Освітньо-наукова програма «Технології машинобудування»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ **Юрій ПЕТРАКОВ**

«10» вересня 2018 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту
Артюшенку Віктору Владиславовичу**

1. Тема дисертації «Використання алгоритмів штучного інтелекту в дослідженнях технологій машинобудування», науковий керівник дисертації Фролов Володимир Костянтинович, кандидат технічних наук, доцент, затверджені наказом по університету від «04» березня 2020 р. № 800-с

2. Термін подання студентом дисертації

15.05.2020

3. Об'єкт дослідження:

системи штучного інтелекту

4. Предмет дослідження:

властивості систем штучного інтелекту, які використовуються для дослідження та управління технологічними процесами.

5. Перелік завдань, які потрібно розробити:

провести літературний пошук за темами: «системи штучного інтелекту», «алгоритми машинного навчання», «ортогональне фрезо-токарне оброблення», спланувати та провести експеримент, провести аналіз результатів експерименту, розробити програмне забезпечення з

використанням алгоритмів штучного інтелекту, розробити систему адаптивного управління режимами різання та моніторингу.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу

Презентація PowerPoint

7. Орієнтовний перелік публікацій

1-2 статті, 1-2 патенти

8. Дата видачі завдання

30.09.2018

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Проведення літературного пошуку за темами: «системи штучного інтелекту», «алгоритми машинного навчання», «ортогональне фрезо-токарне оброблення»	1.02.2019	виконано
2	Планування експерименту	1.05.2019	виконано
3	Проведення експерименту	1.08.2019	виконано
4	Аналіз результатів експерименту та підготовка даних до навчання алгоритмів	1.10.2019	виконано
5	Розробка програмного забезпечення з використанням алгоритмів штучного інтелекту	1.01.2020	виконано
6	Валідування отриманих емпіричних моделей	1.03.2020	виконано
7	Розробка схеми адаптивного управління режимами різання та моніторингу	1.04.2020	виконано
8	Оформлення дисертації	15.05.2020	виконано

Студент

Віктор АРТЮШЕНКО

Науковий керівник

Володимир ФРОЛОВ

АНОТАЦІЯ

Робота містить: 90 аркушів, 59 ілюстрацій, 22 таблиці, 2 додатки. При написанні роботи використано 69 літературних джерел.

Об'єкт дослідження – системи штучного інтелекту

Предмет дослідження – властивості систем штучного інтелекту, які використовуються для дослідження та управління технологічними процесами.

Мета роботи – підвищення контрольованості нових технологічних процесів оброблення на прикладі процесу ортогонального фрезо-токарного оброблення при зменшених витратах за рахунок оптимізації створення моделей процесів оброблення та застосування гнучких систем адаптивного управління та моніторингу.

Методи дослідження: ідеалізація, формалізація, аналіз, синтез, індукція, комп'ютерне математичне моделювання.

Наукова новизна одержаних результатів

1. Запропоновано використовувати системи штучного інтелекту для підвищення контрольованості процесу фрезо-токарного оброблення.
2. Автоматизовано створення емпіричних моделей процесів оброблення зі складною кінематикою за допомогою штучного інтелекту.
3. Розроблене програмне забезпечення зі штучним інтелектом для проведення міждисциплінарних досліджень процесів оброблення
4. Запропонована система штучного інтелекту для адаптивного управління режимами різання та моніторингу стану інструмента.

Практичне значення одержаних результатів

1. Розроблене програмне забезпечення надає можливість автоматизувати створення емпіричних моделей процесів оброблення, що дозволяє заощадити час та собівартість за рахунок зменшення кількості експериментів в порівнянні з традиційними методами моделювання. Це також

виключає виникнення помилок за рахунок людського фактору. Програмне забезпечення може бути застосовано для будь-якого технологічного процесу без зміни вихідного коду за рахунок здатності алгоритмів до самонавчання. За результатами роботи програми користувач може обрати оптимальну емпіричну модель та використовувати її для симуляцій.

2. Запропонована система штучного інтелекту для адаптивного управління режимами різання та моніторингу стану інструмента, на відміну від традиційних систем, є гнучкою та може пристосовуватися до змін умов оброблення та бути застосована до будь-якого процесу оброблення. Її застосування дозволить забезпечити точність оброблення або збільшити продуктивність процесу оброблення за рахунок оптимізації режимів різання. Ця система є частиною концепцій «Індустрія 4.0» та «Agile Manufacturing».

Пропозиції щодо можливих напрямків продовження досліджень

Подальші дослідження доцільно проводити в напрямку поповнення бази даних розробленої системи штучного інтелекту та створення апаратного забезпечення для цієї системи.

Ключові слова: ОРТОГОНАЛЬНЕ ТОКАРНО-ФРЕЗЕРНЕ ОБРОБЛЕННЯ, МАШИННЕ НАВЧАННЯ, ОПТИМІЗАЦІЯ, ШТУЧНИЙ ІНТЕЛЕКТ, АДАПТИВНЕ УПРАВЛІННЯ.

Під час навчання загалом опубліковано 12 тез та матеріалів конференцій (в тому числі 5 – міжнародних), отримано 1 свідоцтво на право власності на твір (комп'ютерна програма, яка впроваджена у навчальний процес) та 6 патентів України на корисну модель. Зокрема, під час роботи над магістерською дисертацією опубліковано 4 матеріали конференцій (в тому числі 3 – міжнародних), отримано 4 патенти України на корисну модель..

ABSTRACT

The thesis contains 90 pages, 59 illustrations, 22 tables, 2 appendices. 69 literary sources were used in the paper.

The object of research – artificial intelligence systems.

The subject of research – properties of artificial intelligence systems used for research and manufacture process control.

The purpose of thesis – to increase of controllability of new machining processes on an example of turn-milling at the reduced expenses due to optimization of creation of process models and application of flexible systems of adaptive control and monitoring.

Methods of research: idealization, formalization, analysis, synthesis, induction, computer mathematical modeling.

Scientific novelty of the received results

1. It is proposed to use artificial intelligence systems to increase the controllability of turn-milling process.
2. Creation of empirical models of machining processes with complex kinematics were automatized using artificial intelligence.
3. Software with artificial intelligence for interdisciplinary research of machining processes was developed
4. The system of artificial intelligence for adaptive control of cutting modes and tool condition monitoring is offered.

The practical value of the obtained results

1. The developed software allows automatizing creation of empirical models of machining processes, which saves time and manufacturing cost by reducing the required number of experiments, compared with traditional modeling methods. This also eliminates errors by the calculations due to the human factor. The software can be used for any technological process without changing the source code due to the

ability of algorithms to self-learning. Based on the results of the program, the user can choose the optimal empirical model and use it for simulations.

2. The proposed scheme of artificial intelligence for adaptive control of cutting modes and tool condition monitoring compared with the traditional systems is flexible and can adapt to changes in machining conditions and can be applied to any machining process. Their application will allow to ensure the accuracy of processing or to increase the productivity of the processing due optimizing the cutting modes. This system is part of the "Industry 4.0" and "Agile Manufacturing" concepts.

Proposals for possible directions of research continuation

It is reasonably to do further researches in the direction of adding data to the database of the developed artificial intelligence system and creating hardware for this system.

Keywords: ORTHOGONAL TURN-MILLING, MACHINE LEARNING OPTIMIZATION, ARTIFICIAL INTELLIGENCE, ADAPTIVE CONTROL.

During the studying generally 12 materials of reports at conferences were published (including 5 at international ones), 1 certificate of ownership for a work (computer program which was implemented into the educational process) and 6 patents of Ukraine for utility model were received. Among them during work on the master's thesis 4 materials of reports at conferences (including 3 at international ones) were published, 4 patents of Ukraine for utility model were received.

Inhaltsverzeichnis

Einleitung.....	1
1 Der Einsatz künstlicher Intelligenz in der Weltpraxis	5
1.1 Begriffe.....	5
1.2 Anwendungen im Maschinenbau	10
2 Experimenteller Teil	17
2.1 Versuchsplanung	17
2.2 Versuchsaufbau	18
2.3 Versuchsdurchführung.....	22
2.4 Messung.....	23
3 Analyse und Bearbeitung der Experimentenergebnisse.....	28
3.1 Datenbewertung	28
3.2 Vorbereitung.....	29
3.2.1 Funktions- und Merkmalsauswahl	29
3.2.2 Datenreinigung	30
3.2.3 Datensatzerstellung.....	30
3.2.4 Codierung	33
3.2.5 Datenskalisierung.....	33
3.3 Korrelationsprüfung.....	34
3.4 Datenzerteilung	37
4 Entwicklung und Konzipierung von ML-Systemen.....	40
4.1 Vorwahl der Algorithmen nach Aufgabekriterien.....	40
4.2 Regressionsalgorithmen	42
4.2.1 Lineare Regression mithilfe des Konjugierte-Gradienten-Verfahrens	42
4.2.2 Polynome Regression mithilfe des Konjugierte-Gradienten-Verfahrens	47
4.2.3 Support Vector Regression (SVR).....	53
4.3 Statistisches Verfahren für Regressionsaufgabe	56
4.4 Auswahl der Regressionsalgorithmen.....	59
4.5 Klassifikationsalgorithmen.....	64
4.5.1 Logistische Regression mithilfe des Konjugierte-Gradienten-Verfahrens	64
4.5.2 Neuronale Netzwerke	70
4.5.3 Support Vector Machines	73
4.6 Auswahl der Klassifikationsalgorithmen.....	76
4.7 Anwendungsschemen der Algorithmen und Modelle.....	78
4.7.1 Schema zur Optimierung der wissenschaftlichen Forschungen.....	78
4.7.2 Schema zur Steuerung und Überwachung des Drehfräsens in der Produktion	80

Zusammenfassung	83
Literaturverzeichnis.....	85

Einleitung

Im modernen Maschinenbau sind schwer bearbeitbare Werkstoffe eingesetzt. Beispiele für solche Materialien sind CoCrMo-Legierungen, die aufgrund ihrer biomechanischen Eigenschaften in der Medizin häufig zur Herstellung von Prothesen verwendet werden [1].

Hohe Anforderungen an Sauberkeit und Oberflächenqualität sowie mechanische Eigenschaften von CoCrMo-Legierungen erschweren den Prozess ihrer Bearbeitung erheblich. Der schnelle Verschleiß des Schneidwerkzeugs und der komplexe Prozess der Reinigung der behandelten Oberflächen erhöhen die Kosten für die Herstellung von Prothesen und dementsprechend die Kosten für das Endprodukt. Daher besteht die Notwendigkeit, diesen Prozess zu optimieren, um die Verschleißrate von Schneidwerkzeugen zu verringern und qualitativ hochwertige bearbeitete Oberflächen zu erhalten.

Um diese Ziele zu erreichen, wird in dieser Arbeit vorgeschlagen, anstelle des konventionellen Drehens orthogonales Drehfräsen zu verwenden. Zu den Vorteilen dieses Verfahrens gehören ein produktiverer Materialabtrag, Spanzerkleinerung, weniger Werkzeugdruck auf das Werkstück, eine geringere Verschleißrate aufgrund der Lastverteilung auf mehrere Wendeschneidplatten und eine geringere Rauheit der bearbeiteten Oberflächen [2]. Diese Bearbeitungstechnologie wurde bereits erfolgreich als Alternative zum Drehen für die Bearbeitung von gehärtetem Stahl und Aluminium eingesetzt [3].

Da Kurvenfläche auf einer Drehfräsmaschine bearbeitet werden und asymmetrische Teile erhalten werden können, ist es auch möglich, die Anzahl der zur Herstellung von Prothesen erforderlichen Maschinen zu verringern.

Der Drehfräsenprozess wurde aufgrund der relativen Neuheit des Verfahrens selbst sowie der komplexen Kinematik und dynamischen Bearbeitungsbedingungen noch nicht ausreichend untersucht. Also nur wenige Informationen zu optimalen Zerspanungswerten bei der Bearbeitung verschiedener Materialien vorliegen. Bestehende Formeln zur Berechnung der Zerspanungswerte beim Drehfräsen können nur für Vorberechnungen eingesetzt werden [4]. Aus diesem Grund wird dieses Verfahren in der Produktion nicht oft eingesetzt. Weitere Forschungen sind erforderlich, um diesen Prozess kontrollierter zu gestalten.

In den wissenschaftlichen Forschungen gelten die Experimente als die Kernelemente bei der Identifizierung der Eigenschaften von untersuchten Objekten und bei der Überprüfung der Gültigkeit von Hypothesen. In der Fertigungstechnik sind ihre Vorbereitung und Durchführung mit dem Zeit- und Kostenaufwand verbunden und deshalb besteht die Optimierungssaufgabe in der Verringerung der benötigten Versuchsanzahl. Auf Basis der Versuchsergebnisse können die Modelle erstellt werden, die in der Zukunft in den weiteren Forschungen oder in

der Produktion angewendet werden. Die Voraussetzung für die Modellerstellung ist die vorhandenen empirischen Daten und Kenntnisse über die physikalischen Prinzipien des Prozesses.

In der Fertigungstechnik sind ausreichende Informationen zum System nicht immer verfügbar d.h. keine angemessenen und vollständigen theoretischen Kenntnisse bzw. Modelle vorliegen. In diesen Fällen verwenden Wissenschaftler seit langen statistischen Techniken, um ihre Arbeit zu unterstützen [5]. Statistik bzw. statistische Modellierung wurde bevor Computer erfunden und aus Mathematik herkommt. Zurzeit ist mit der Computertechnikentwicklung noch eine Technik entstanden, die diese Aufgabe erfüllen könnte nämlich „Machine Learning“ (ML).

Seit Jahr 2015 ist der Anstieg des Interesses am Thema „Machine Learning“ in unterschiedlichen Bereichen durch das Wachstum der Forschungen (Abbildung 1) und den Investitionsumfang bemerkbar (Abbildung 2). Dazu wurden billig verfügbare Rechenressourcen, das Wachstum der Rechenleistung, die Verfügbarkeit großer Datenmengen in Verbindung mit der Verbesserung der Algorithmen geführt. Als Folge werden Technologien der Künstlichen Intelligenz (KI) heute in einer Vielzahl von wissenschaftlichen Forschungsbereichen eingesetzt [5].

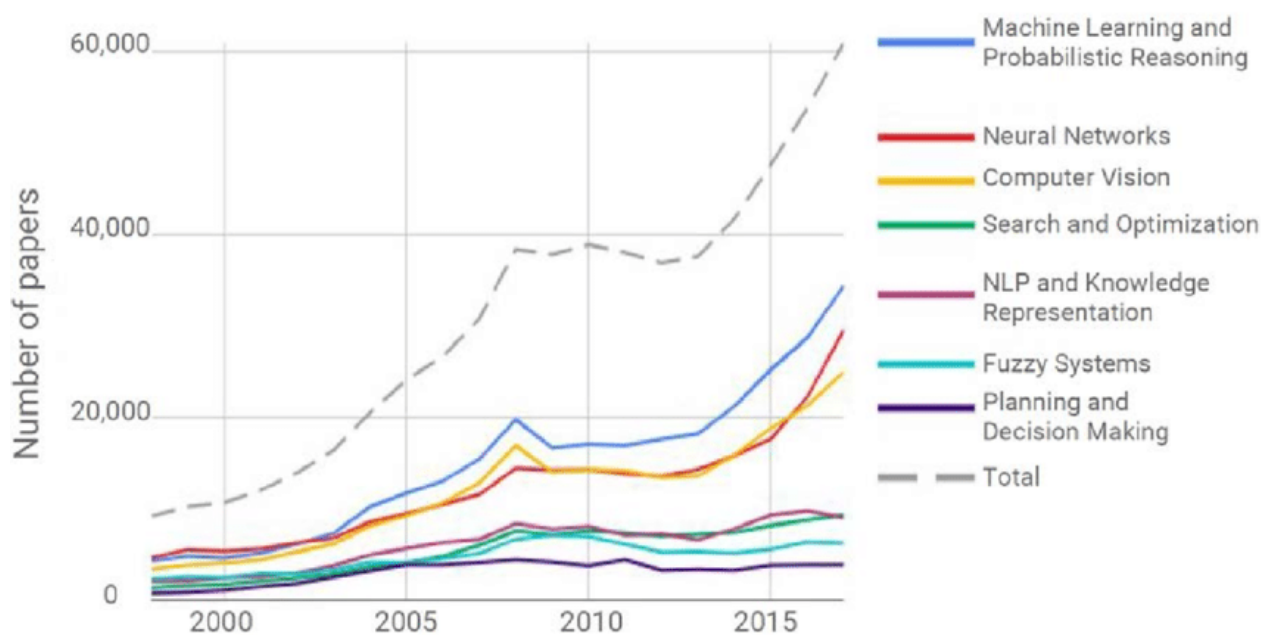


Abbildung 1: Anzahl der Jährlich veröffentlichte KI-Artikel in Scopus nach Unterkategorie (1998-2017) [6]

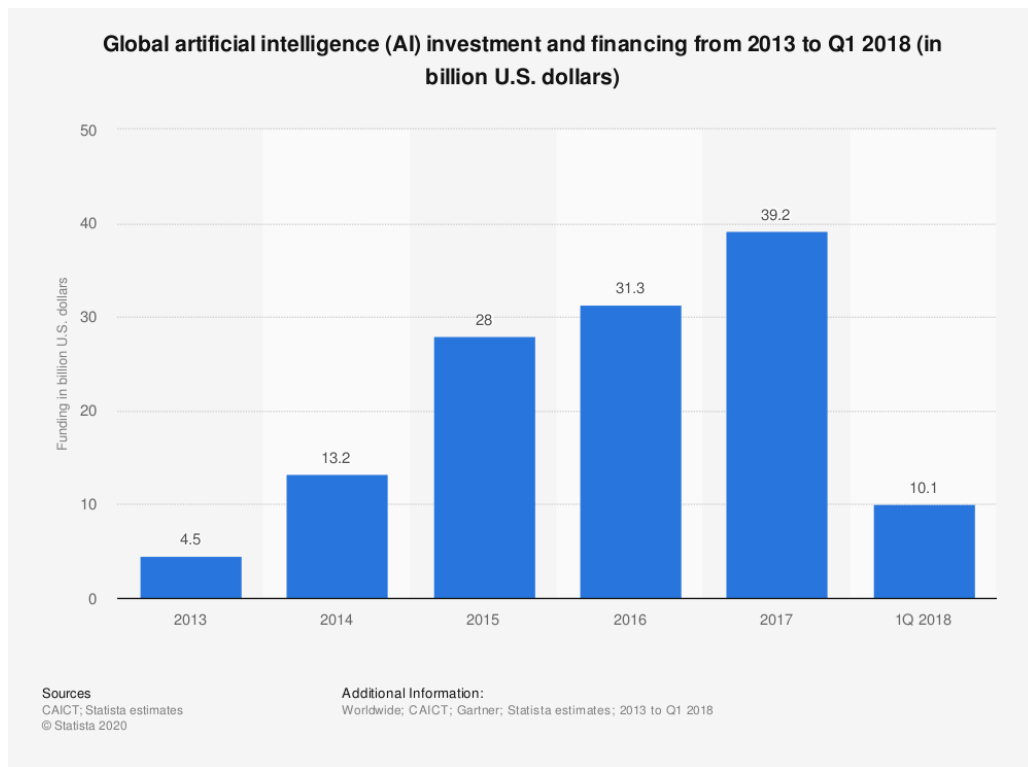


Abbildung 2: Globale Investitionen und Finanzierungen für künstliche Intelligenz von 2013 bis zum ersten Quartal 2018 (in Milliarden US-Dollar) [7]

Der Schwerpunkt von Machine Learning liegt auf der automatischen Erstellung empirischer Modelle aus Daten mit Hilfe der Computeralgorithmen. Die dabei benutzte Optimierungsverfahren und größere Ausnutzung der Rechenleistung ermöglichen Modellierung zu beschleunigen und weniger Daten dabei zu brauchen (Abbildung 3) d.h. Kosten- und Zeitaufwand zu vermindern.

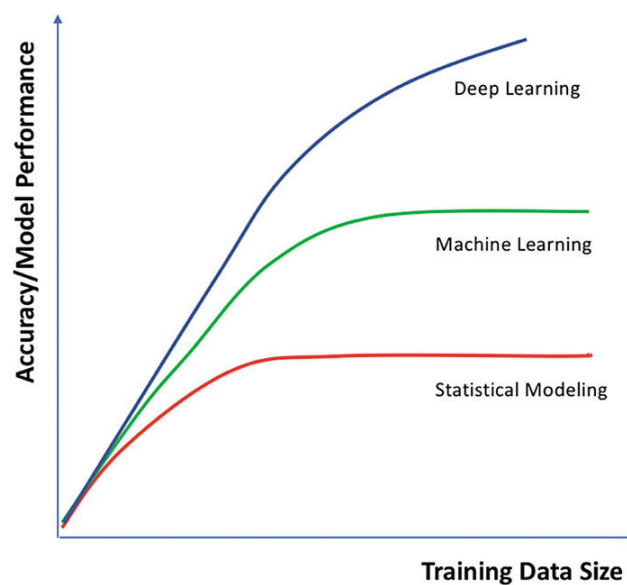


Abbildung 3: Modellleistung mit Datengröße für drei Felder [8]

Darüber hinaus ermöglicht dieses neues Verfahren Muster und Anomalien in Datensätzen zu erkennen, die weitaus größer und unordentlicher sind, als Menschen damit umgehen können oder überraschende, nicht intuitive Muster. Diese potenziellen Vorteile bilden die Motivation für die Analyse der Anwendbarkeit dieser neuen Methode für Forschungsanwendungen im Bereich Fertigungstechnik.

1 Der Einsatz künstlicher Intelligenz in der Weltpraxis

Die Untersuchungsgegenstände dieser Arbeit sind Techniken der künstlichen Intelligenz, die zum Teilbereich Machine Learning gehören. Prognosen zufolge wird der Einsatz dieser Technologien zunehmen (Abbildung 1-1), was auch durch Investitionen in die Forschungen in diesem Bereich bestätigt wird (Abbildung 2). Für die Erfüllung der Aufgabe werden daher Kenntnisse und Verständnis für grundlegende Konzepte benötigt. Infolgedessen wurde im Rahmen dieser Arbeit entsprechende Analyse vom Stand der Technik durchgeführt, die Begriffsklärung und Übersicht der realisierten Projekte in der Fertigungstechnik umfasst.

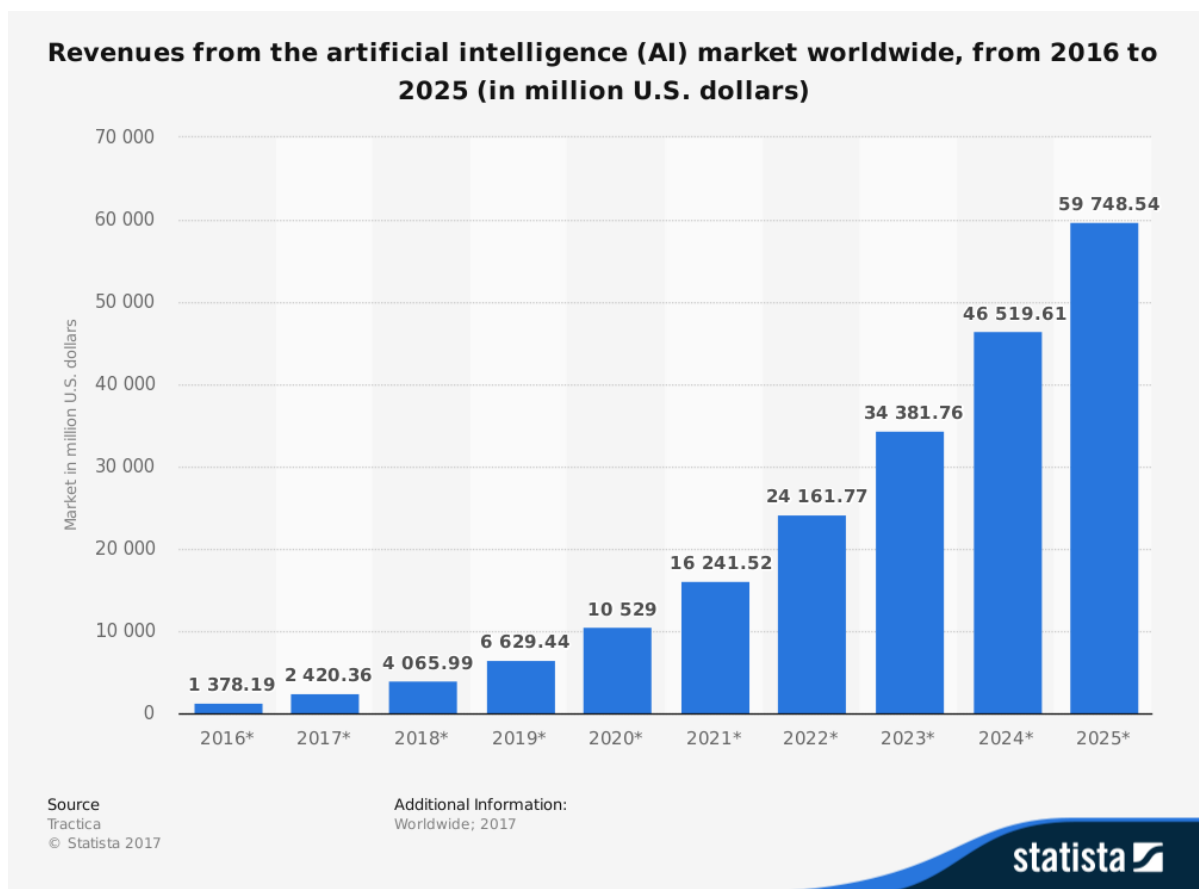


Abbildung 1-1: Einnahmen aus dem weltweiten Markt für künstliche Intelligenz von 2016 bis 2025 (in Millionen US-Dollar) [9]

1.1 Begriffe

Es gibt viele Definitionen von KI. In dieser Arbeit wurden 8 von ihnen analysiert und festgestellt, dass sie in 4 Gruppen unterteilt werden können (Abbildung 1-2) [10].

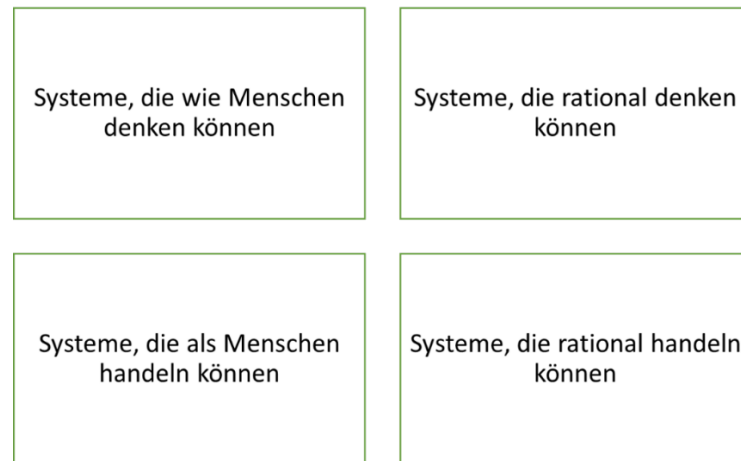


Abbildung 1-2: Einordnung der KI-Begriffe in vier Gruppen

In den Gruppen links (Abbildung 1-2) wird der Erfolg anhand der menschlichen Aktivität und in Gruppen rechts (Abbildung 1-2) anhand des idealen Konzepts der Intelligenz (Rationalität) gemessen. Die Betrachtung von KI als rationales System ist eine idealisierte Darstellung, die aufgrund der Komplexität der Implementierung und des fehlenden objektiven Bewertungsverfahrens keine praktische Anwendung findet [10].

Deswegen wird in dieser Arbeit weiter KI, als menschenbezogenes System angenommen. Aus dieser Sicht kann KI folgendermaßen definiert werden: Theorie und Entwicklung von Computersystemen, die Aufgaben ausführen können, die normalerweise menschliche Intelligenz erfordern, wie visuelle Wahrnehmung, Spracherkennung, Entscheidungsfindung und Übersetzung zwischen Sprachen [11].

Das Gebiet der künstlichen Intelligenz umfasst mehrere Bereiche (Abbildung 1-3):

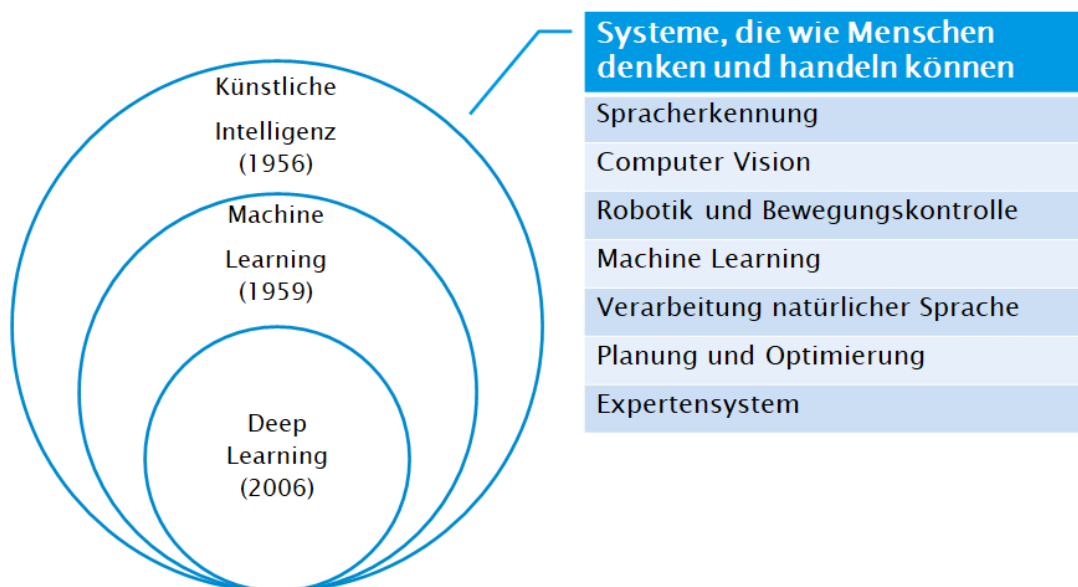


Abbildung 1-3: Gliederung der KI-Bereiche

Zur Analyse der Versuchsergebnisse bei den Forschungen von Drehfräsen wurden in dieser Arbeit Machine Learning eingesetzt.

Machine Learning (ML) ist ein Teilgebiet der Informatik, in dem Computer lernbar sind, ohne explizit programmiert zu werden [12].

Die klassische Statistik ist das Kernstück des maschinellen Lernens und viele Algorithmen basieren auf statistischen Gleichungen. Beim ML werden statistische Modelle verwendet, um Muster zu erkennen und die Leistung basierend auf den Daten und empirischen Informationen ohne direkte Programmierbefehle zu verbessern. Dies bedeutet nicht, dass die Programmierung nicht dabei verwendet wird, sondern dass die Programmierung als Werkzeug für Datenverarbeitung angewandt wird. Der Unterschied zwischen konventioneller Programmierung und maschinellem Lernen ist auf der Abbildung 1-4 dargestellt:

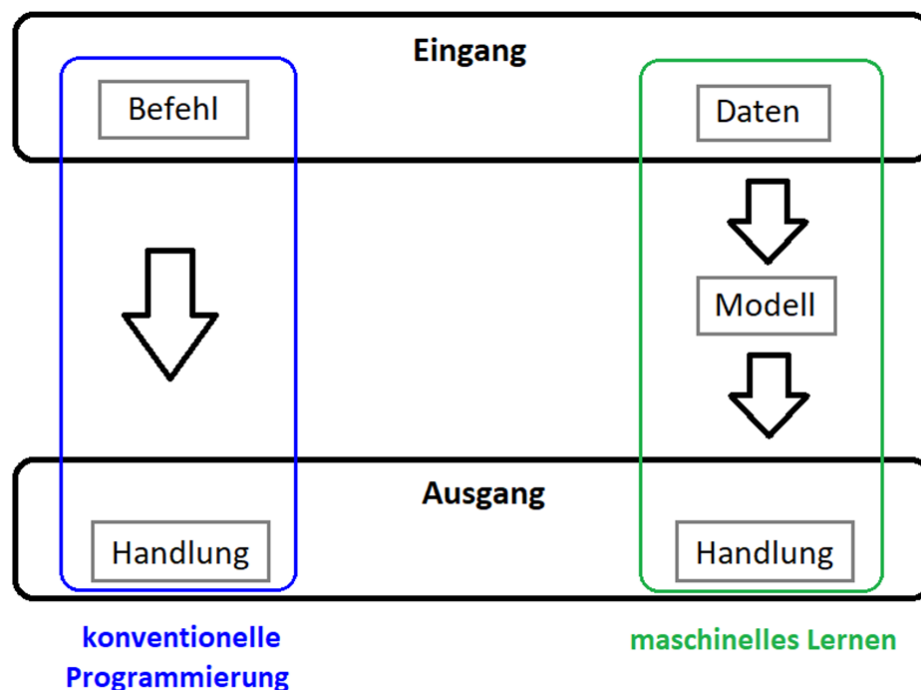


Abbildung 1-4: Differenz zwischen der konventionellen Programmierung und dem maschinellen Lernen

Beim maschinellen Lernen werden Daten eingegeben, ein Algorithmus ausgewählt und seine Hyperparameter (Einstellungen) konfiguriert. Algorithmen suchen durch Versuch und Irrtum (iterativ) nach einer mathematischen Formel (Modell), die bei Anwendung auf einen Eingabedatensatz (als "Trainingsdatensatz" bezeichnet) die gewünschten Ergebnisse liefert. Diese mathematische Formel erzeugt auch die korrekte Ausgabe für die meisten anderen Eingabedaten (die sich von Trainingsdaten unterscheiden), unter der Voraussetzung, dass diese Eingabedaten aus derselben oder einer ähnlichen statistischen Verteilung stammen, aus der die Trainingsdaten erhalten wurden [13]. Solche Daten werden als Testdaten bezeichnet,

da sie zur Verifizierung des resultierenden Modells verwendet werden. Das allgemeine Trainingsschema ist auf der Abbildung 1-5 illustriert:

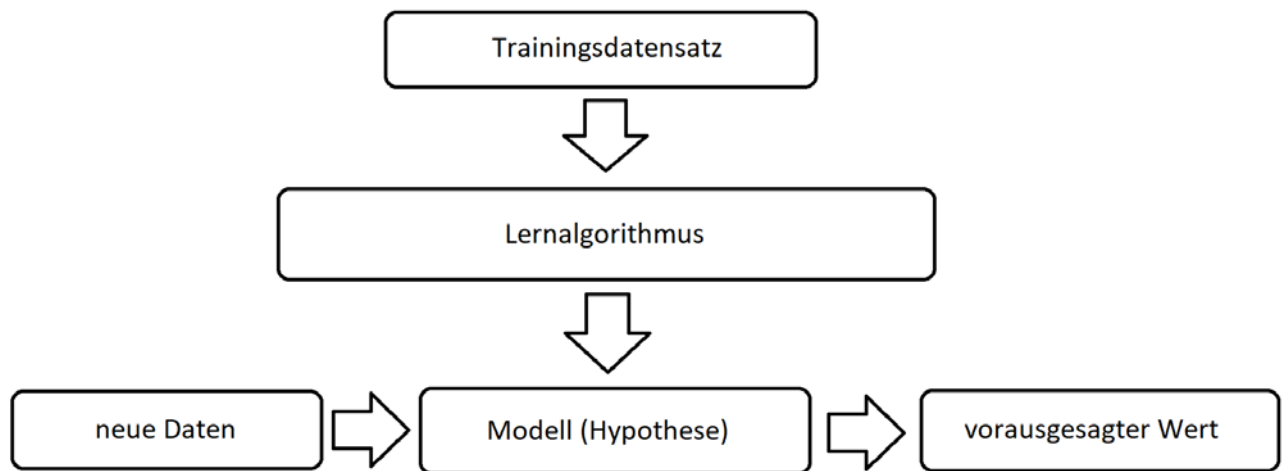


Abbildung 1-5: Verallgemeinerter Lernalgorithmus

Für die Inbetriebnahme des Modells muss gleichzeitig eine zufriedenstellende Genauigkeit mit der Trainings- und Testdaten erreicht werden.

In diesem Prozess verläuft das Lernprozess nicht wie bei Menschen oder Tieren, denn wenn die Eingaben leicht verzerrt werden, wird die Ausgabe wahrscheinlich völlig falsch. D.h. das Wort "Lernen" im Begriff wird nicht wörtlich, sondern in Analogie verwendet [13].

Die praktische Aufgabe in dem ML besteht darin, den richtigen Algorithmus oder eine Kombination von Algorithmen für einen bestimmten Datensatz auszuwählen. Alle Algorithmen sind in drei Typen geteilt:

- Überwachtes Lernen
- Unüberwachtes Lernen
- Verstärkungslernen

In diesem Masterprojekt wurden überwachte Methoden des maschinellen Lernens verwendet, um die Regressions- und Klassifizierungsmodelle für die Daten des Bearbeitungsprozesses zu erstellen.

Beim überwachten Lernen enthält ein Eingabedatensatz (Trainingssatz) Informationen, die in Testbeispielen nicht vorhanden sind. In diesem Fall zielen die während des Trainings gesammelten Erfahrungen darauf ab, die fehlenden Informationen für die Testdaten vorherzusagen.

Der Datensatz besteht aus Beispielen, die eine Sammlung von Informationen aus der Umgebung enthalten. Beim überwachten Lernen besteht ein Beispiel aus den Merkmalen (mit X bezeichnet), die das Beispiel charakterisieren, und den korrekten Ausgabewert (mit Y bezeichnet).

bezeichnet). Beispielsweise wird fürs Drehfräsen eine Bearbeitungsoperation als Beispiel, Zerspanungsdaten sind als die Merkmale (X) und die Temperatur in der Zerspanungszone (für das Regressionsmodell) oder der Werkzeugzustand Werkzeugs (für das Klassifizierungsmodell) sind als korrekte Ausgabewert (Y).

Der Algorithmus empfängt die Merkmale X und die richtigen Antworten Y am Eingang, decodiert Muster zwischen den Daten und erstellt ein Modell, das diese Muster mit den neuen Daten reproduzieren kann. Wenn er die richtige Antwort erhält, arbeitet er umgekehrt, um die Beziehungen zwischen Eingabe und Ausgabe zu bestimmen. Das Schema des überwachten Lernens ist auf der Abbildung 1-6 geschildert:

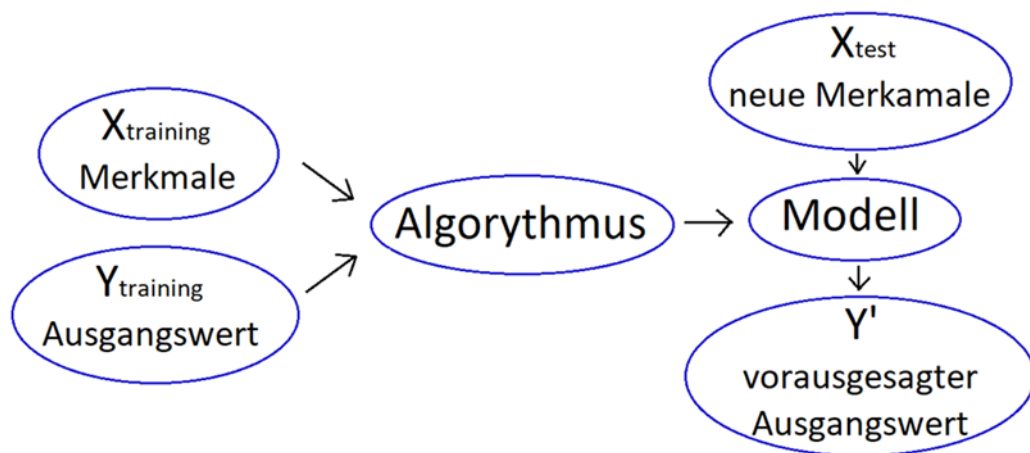


Abbildung 1-6: Schema des Lernprozessverlaufs beim überwachten maschinellen Lernen

Deep Learning (DL) ist ein neues ML-Gebiet, das mit dem Ziel eingeführt wurde, maschinelles Lernen einem seiner ursprünglichen Ziele näher zu bringen: Künstliche Intelligenz. Beim Deep Learning geht es darum, mehrere Ebenen der Repräsentation und Abstraktion zu lernen, die dazu beitragen, Daten wie Bilder, Ton und Text zu erkennen und zu verstehen [14].

Deep Learning ist auf künstliche neuronale Netze spezialisiert. Im Gegensatz zu neuronalen Netzen, die beim maschinellen Lernen benutzt werden, werden hier andere Netzdesigns und andere Parameterinitialisierungen verwendet.

Die Hauptvorteile tiefer neuronaler Netzwerke gegenüber allen Algorithmen für maschinelles Lernen:

- Die Leistung tiefer neuronaler Netzwerke wächst mit zunehmender Eingangsdatenmenge weiter, während Algorithmen für maschinelles Lernen irgendwann Spitzenleistung erreichen (Abbildung 1-7)

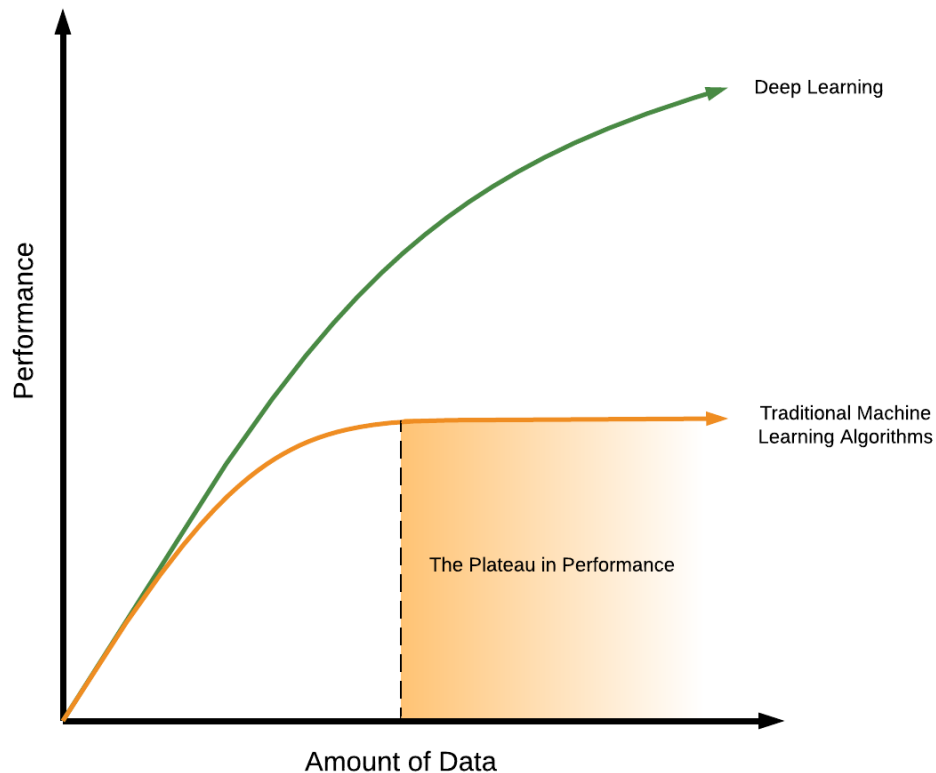


Abbildung 1-7: Abhängigkeit der Leistung von Datenmenge bei Lernalgorithmen [15]

- Genauigkeit der Erkennung ist wesentlich höher
- Relevante Merkmale automatisch extrahiert, was besonders wichtig bei der Bild-, Ton- und Videoerkennung ist. Beim Maschinellen Lernen wird das manuell erledigt.

Die Anwendungsvoraussetzungen des Deep Learning sind hohe Rechenleistung und große Datenmenge (mehr als 1000 Beispiele) [14].

In dieser Masterarbeit sind keine ausreichende Datenmenge vorhanden und es gibt keinen Bedarf eine Bild-, Ton- oder Videoerkennung durchzuführen. Aus diesem Grund werden die Techniken von Deep Learning in dieser Arbeit nicht betrachtet und eingesetzt.

1.2 Anwendungen im Maschinenbau

ML wird schon häufig verwendet, um kritische Entscheidungen in der Medizin, auf den Finanzmärkten und im Energiesektor zu unterstützen. Obwohl ML zunehmend Eingang in der Maschinenbauindustrie findet, ist die Produktionstechnik zurzeit in einer frühen Phase der ML-Anwendung [16]. Das liegt daran, dass die Natur von Fertigungssystemen immer komplexeren, dynamischeren und manchmal sogar chaotischeren Verhaltensweisen ausgesetzt ist [17]

Bei einem hohen Entwicklungsstand und großen Produktionsumfang kann bereits ein geringer Kosten- oder Prozesszeiteinsparung einen Vorteil für Unternehmen auf dem Markt schaffen

und das Einkommen steigern. Anwendung der ML-Techniken ermöglicht solcher Gewinn durch die Prozessoptimierung und die Produktqualitätserhöhung zu erzielen.

Traditionelle Methoden, die auf der Modellierung von Ursache-Wirkungs-Beziehungen basieren, stoßen aufgrund der schnell wachsenden Komplexität und hohen Dimensionalität moderner Fertigungsprogramme an ihre Grenzen [18]. ML-Techniken sind imstande einige der heutigen großen Herausforderungen komplexer Fertigungssysteme anzunehmen, weil sie dabei folgende Vorteile bieten:

- die Fähigkeit, hochdimensionale Probleme und Daten zu behandeln
- die Fähigkeit, früher unbekanntes (implizites) Wissen zu entdecken und implizite Beziehungen in Datensätzen zu identifizieren
- Echtzeitverarbeitung
- Umgang mit verschiedenen Arten von Daten (numerisch, nominal, Text und Bilder)
- verbesserte Benutzerfreundlichkeit der Anwendung von Algorithmen
- ermöglicht es, die Anzahl der benötigten Proben in bestimmten Fällen zu reduzieren

Es gibt aber einige Schlüsselprobleme, die noch Anlass zur Unsicherheit sind und manchmal Unternehmen auch davon zurückhalten, sinnvolle Investitionen in ML abzuwickeln. Dazu gehören unklare rechtliche Aspekte (z.B. Wer ist verantwortlich, wenn Menschen Entscheidungen an Maschinen delegieren?) und zeitaufwändige, schwierige Datenaufbereitung [16].

Obwohl ML in den meisten Fällen das Extrahieren von Wissen ermöglicht und bessere Ergebnisse erzielt als die meisten herkömmlichen Methoden, müssen bestimmte Aspekte noch berücksichtigt werden [17]. Die Erfassung der relevanten Daten ist häufige Herausforderung in der Fertigung, da die Verfügbarkeit, Qualität und Zusammensetzung der vorliegenden Herstellungsdaten einen starken Einfluss auf die Leistung von ML-Algorithmen haben (z.B. hochdimensionale Daten können große Menge von irrelevanten und redundanten Informationen enthalten) [19].

Eine weitere Herausforderung ist die Interpretation der Ergebnisse. Es muss berücksichtigt werden, dass nicht nur das Format oder die Darstellung der Ausgabe für die Interpretation wichtig ist, sondern auch die Spezifikationen des gewählten Algorithmus und die Parametereinstellungen [17].

Die allgemeine Fähigkeit der ML-Algorithmen, Ergebnisse in einer Fertigungsumgebung zu erzielen, wurde jedoch erfolgreich bewiesen [17].

Während der Literaturrecherche wurden Anwendungsbeispiele des maschinellen Lernens in der Produktionstechnik gefunden. Diese Studien können nach Forschungsrichtungen in folgende Gruppen unterteilt werden:

- Bildverarbeitung
- Adaptive Steuerung
- Überwachung des Werkzeug- / Maschinenzustands
- Qualitätsüberwachung in der Fertigung
- Prozessoptimierung und Automatisierung

ML-basierte Bildanalysesysteme werden dort eingesetzt, wo herkömmliche Bildverarbeitungssysteme an ihre Grenzen stoßen und menschliches Urteilsvermögen und Sehvermögen die beste Lösung ist. Solche Systeme bieten Stand der Technik Lösung und das Potenzial für eine zuverlässige, automatisierte Verifizierung mit sehr hohen Erkennungswerten. Neue Produkte oder unbekannte Merkmale können ohne großen Aufwand erlernt und erkannt werden, was zu erheblich kürzeren Entwicklungs- und Produkteinführungszeiten führt. Die Modellierung erfordert dabei keine zusätzliche Softwareentwicklung und kein Verständnis der Algorithmen [16].

Für die ML-Bildverarbeitung können zur Visualisierung von Prozessen alle Arten von Sensoren verwendet werden, einschließlich 2D-, 3D-, Ultraschall und Shape from Shading Technik [16].

Wichtige einschränkende Faktoren können dabei die Zugänglichkeit der erforderlichen Trainingsdaten, die Sicherheitsanfälligkeit des ML-Systems gegenüber Manipulationen oder Sabotage und die Bildauflösung der Bilder sein [16].

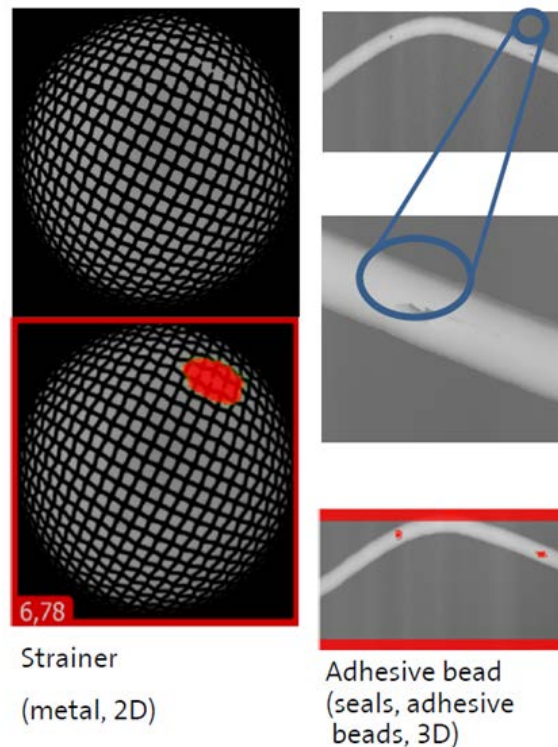


Abbildung 1-8: Beurteilung von Oberflächenstrukturen in 2D und in 3D bei i-mation [16]

Diese Techniken werden bereits für die Erkennung von Qualitätsfehlern (z.B. Risse) direkt an Aluminium-Stanzteilen in Echtzeit anhand von Bildern [20], für die Online-Materialfehlererkennung durch Bildverarbeitung beim selektiven Laserschmelzen (SLM) von Metallpulvern [21], für automatische Qualitätsprüfung von gedruckten Werkstücken beim Fused Deposition Modeling(FDM) durch Bildanalyse der Zwischenzustände [22] und für automatische Erkennung von Mikrostrukturen mit dendritischen Morphologien auf den Bilder eingesetzt [23].

Technische Systeme mit einem Verhalten, das von zahlreichen Variablen und unbekannten Beziehungen beeinflusst wird, sind mit physikalischen Formeln schwer zu modellieren. Das behindert häufig die Prozessoptimierung. In diesem Fall können ML-Methoden hilfreich sein. Sie lernen das Verhalten des Systems und können anschließend Vorhersagen über den Prozess machen und die Prozessparameter verändern, was als adaptive Steuerung bezeichnet wird [16].

Auf dem ML basierte adaptive Steuerung hat schon die Anwendungen in der Fertigungstechnik gefunden. Zu den Beispielen gehören die Anpassung der Vorschubgeschwindigkeit und der Spindeldrehzahl zur Verringerung des Flankenverschleißes beim Fräsen [24], adaptive Regulierung der Spindeldrehzahl und der Vorschubgeschwindigkeit für die Maximierung der Materialabtragsrate beim Drehen [25], Entwicklung eines adaptiven Steuerungssystems für die Gewährleistung der erfordernten Oberflächenrauheit während des Drehvorgangs anhand den

gemessenen Werkzeugvibrationen [26] und Steuerung von Freiformabscheidungs- oder Verbindungsprozessen durch Anpassungen der Prozesssteuerungsparameter mit vorhandenen Echtzeit-Objekteigenschaftsdaten [27].

Die Überwachung von Bearbeitungsvorgängen beiträgt erheblich zur Automatisierung des Herstellungsprozesses, die die Produktivität jedes Fertigungssystems durch Minimierung der Ausfallzeiten und gleichzeitige Beibehaltung von hohen Qualitätsstandards erhöht [28]. Ein wesentlicher Teil der Maschinenstillstandzeiten ist eine Folge eines Werkzeugausfalls [29]. Infolgedessen hat das Interesse an Tool Condition Monitoring (TCM) erheblich zugenommen, und es wurden dazu umfangreiche multidisziplinäre Forschungsarbeiten durchgeführt, um Werkzeugfehler oder -brüche vor dem Auftreten eines tatsächlichen Ereignisses zu erkennen [30].

Bei der Untersuchung der Literatur zu TCM-Systemen wurde herausgefunden, dass neue Varianten solcher Systeme mit Hilfe der ML-Techniken realisiert wurden. Besonders werden sie beim Fräsen eingesetzt, da ein Fräsprozess aufgrund des Oberflächenkontakts der Werkzeuge mit dem Werkstück im Allgemeinen als komplizierter angesehen wird. ML-Systeme können Werkzeugbrüchen anhand von Daten über die Schnittkräfte und den Stromverbrauch [29] oder über Bearbeitungsparameter [31] in der Echtzeit im Voraus erkennen. Es ist auch möglich Werkzeugzustand anhand von dem Schnittkraftsignal, dem Stromsignal der Spindel und Tonsignal des Bearbeitungsprozesses [28] oder von dem Vibrationssignal [32] zu überwachen.

Eine weitere Möglichkeit zur Automatisierung von Produktionsprozessen ist die automatische Online-Produktqualitätskontrolle während der Fertigung. Um diesen Prozess zu implementieren, ist es sinnvoll, maschinelles Lernen zu verwenden, da Produkte normalerweise viele verschiedene Merkmale und Fälle aufweisen, die erkannt werden müssen. Ein weiterer Grund für die Verwendung von ML ist die Fähigkeit, schnell umzulernen, was für moderne Industrie mit einer breiten Produktpalette wichtig ist.

Als Beispiele kann man Qualitätsüberwachung von Radscheiben [33], Wälzlagern [34], Windkraftwerken [35] und der Schweißqualität [36] angeben.

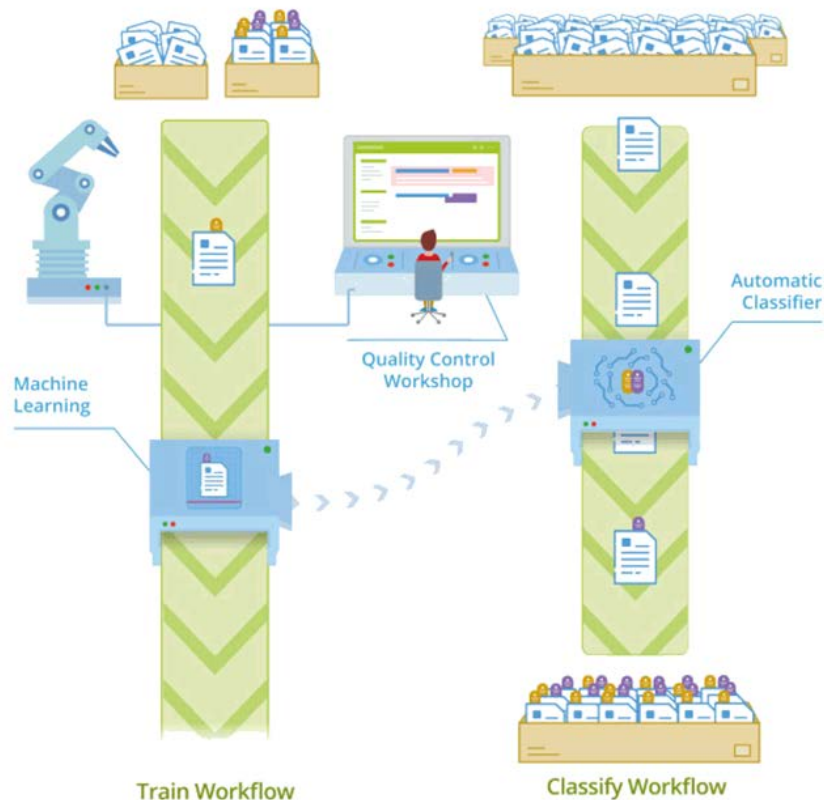


Abbildung 1-9: Allgemeines Qualitätskontrollschema bei Antidot [37]

Es gibt auch andere Möglichkeiten zur Automatisierung und Optimierung von Produktionsprozessen mittels ML-Algorithmen, wie z.B. intelligente Werkzeugweggenerierung zum Fräsen freier Oberflächen für Erzielung der bestmöglichen Oberflächenqualität der bearbeiteten Oberfläche [38], Prozessoptimierung der elektrochemischen Bearbeitung (ECM) [39], Vorhersagen der Schnittwerte beim Fräsen [40] usw.

Die Forschungsprozesse können auch optimiert werden, da ML ermöglicht, die Anzahl der benötigten Proben in bestimmten Fällen zu reduzieren [41] und Probleme wie die Variabilität von Beobachtern besser als mit anderen Methoden zu überwinden [42].

Aus den Ergebnissen dieser Literaturrecherche kann man schließen, dass der ML-Einsatz in der Fertigungstechnik in erster Linie darauf abzielt, Geräteausfälle zu reduzieren, die Produktqualität zu verbessern und die Dauer von Produktionsprozessen zu verkürzen.

Trotzdem werden Systeme, die auf Algorithmen für maschinelles Lernen basieren, für fertigungstechnische Systeme kaum verwendet. ML-Anwendung für Fertigungstechnik ist zurzeit in einer frühen Phase. Für den Prozess von Drehfräsen wurden ML-Techniken früher nicht verwendet.

Daher wird in dieser Arbeit vorgeschlagen, ML-Techniken für weitere Untersuchungen des Drehfräsens zu verwenden. Im Rahmen dieser Masterarbeit wurden Experimente zur Bearbeitung von CoCrMo-Legierungen mittels des orthogonalen Drehfräsens durchgeführt, verschiedene ML-Algorithmen entwickelt und angewendet, um Prozessmodelle zu erstellen. Die resultierenden Modelle haben das Validierungsverfahren bestanden und basierend auf den Ergebnissen wurden die effizientesten Algorithmen ausgewählt. Auf Basis der entwickelten Algorithmen wurden Schema zur Optimierung der wissenschaftlichen Forschungen dieses Verfahrens sowie Schema zur Steuerung und Überwachung des Prozesses in der Produktion vorgeschlagen.

Dadurch wird das orthogonale Drehfräsen kontrollierter. Die entwickelten Systeme, die Technologien für künstliche Intelligenz verwenden, werden es ermöglichen, diese Verarbeitungsmethode in Produktionssystemen zu verbreiten und in das Konzept von "Industrie 4.0" zu integrieren.

2 Experimenteller Teil

Ein Schlüsselement von ML-Projekten sind Daten. Von ihnen hängt in erster Linie die zukünftige Genauigkeit und Leistung der Algorithmen ab. Die Qualität der Daten ist durch die Auswahl und Berücksichtigung aller für das System relevantesten Daten, die Datenerfassung unter realen Bedingungen gekennzeichnet.

Der Vorteil der Fertigungstechnik liegt in der gezielten Datenerfassung, die in anderen Bereichen häufig nicht möglich ist. Somit ist es bereits in der Phase der Datenerfassung möglich, potenziell relevante Parameter und Zielfunktionen auszuwählen.

Um Daten in dieser Arbeit zu sammeln, wurde ein Experiment geplant und durchgeführt. Während des Versuchs wurde der zylindrische Teil aus der CoCrMo-Legierung durch orthogonales Drehfräsen bearbeitet.

2.1 Versuchsplanung

Die Versuchsplanung beginnt mit der Feststellung von Zielgrößen und potenziell relevanten Parametern, die sich auf die Zielgrößen auswirken.

Für die Bearbeitungsprozessen sind die Hauptzielgröße, die den Prozess charakterisieren, die Rauheit, die nach der Oberflächenbehandlung erhalten wird und das Werkzeugverschleiß. Die Oberflächenrauheit kennzeichnet die Qualität des Teils und bestimmt die Auswahl der Bearbeitungsmethode. Das Werkzeugverschleiß kennzeichnet Produktivität und Wirtschaftlichkeit der Bearbeitung. Ziel der Forschung zu Bearbeitungsprozessen in der Fertigungstechnik ist es, die optimalen Schnittbedingungen auszuwählen, um eine geringe Rauheit und geringe Werkzeugverschleißintensität zu erreichen.

Deshalb wurden in dieser Arbeit die Rauheit R_a und die Verschleißtiefe auf zwei Oberflächen der Wendeschneidplatten (WSP) als Zielgröße gewählt.

In bestehenden Studien wurde festgestellt, dass die wichtigsten Einflussparameter auf die Rauheit und den Verschleiß des Werkzeugs beim orthogonalen Drehfräsen die Spindeldrehzahl, die Längsvorschubgeschwindigkeit, die Exzentrizität und die axiale Schnitttiefe sind [43 bis 46].

Es ist unmöglich vorherzusagen, welche Daten und welche Datenmenge für die erforderliche Modellqualität und -genauigkeit benötigt werden [16]. Der ML-Algorithmus erreicht seine maximal mögliche Effizienz, wenn der Datensatz alle möglichen Kombinationen von Merkmalen enthält. Ein Lernmodell sollte in der Regel mindestens zehnmal mehr Datenpunkte aufweisen als die Gesamtzahl der Merkmale. In einigen Fällen ist es unmöglich oder

wirtschaftlich ineffizient, solche Datenmenge zu erfassen. Dabei sollen die zur Verfügung stehende Daten zum Lernen eingesetzt werden [12].

Deswegen wurden nur 19 Experimente in dieser Studie geplant und durchgeführt. Während der Bearbeitung wurde nur ein Parameter geändert, nämlich die axiale Schnitttiefe. Das befriedigt die idealen minimalen Anforderungen an Datenmenge und ermöglicht erste Einschätzung der Funktion von Algorithmen.

Basierend auf den ausgewählten Zielgrößen und Parameter wurden Zielfunktionen bestimmt, mit denen die Algorithmen trainiert werden können. Es ist es Interessant folgende Abhängigkeiten zu untersuchen:

- Verschleißtiefe auf zwei Flächen von der Schnitttiefe
- Oberflächenrauheitsparameter von der Verschleißtiefe der beiden Flächen
- Oberflächenrauheitsparameter von der Schnitttiefe

2.2 Versuchsaufbau

Nachdem der Plan des Experiments entwickelt wurde, ist es notwendig, Ausrüstung und Ausstattung für das Experiment auszuwählen.

Zunächst wurde die Maschine ausgewählt, auf der die Verarbeitung durchgeführt wurde. In diesem Experiment wurde eine DMG MORI CTX Beta 1250 TC Siemens Dreh- und Fräsmaschine verwendet. Auf dieser Maschine ist eine komplexe Bearbeitung von Teilen von sechs Seiten möglich.

Die technischen Daten dieser Maschine sind in Tabelle 2-1 aufgeführt.

Tabelle 2-1: Technische Daten der Maschine DMG MORI CTX beta 1250 TC Siemens [47]

Max. Drehdurchmesser in [mm]	1200
Max. Drehlänge in [mm]	Ø 470
Dreh-Frässpindel / Spindeldrehzahl in [min^{-1}]	12000
Werkzeugmagazin	24
X / Y / Z-Hub Schlitten Fahrständer in [mm]	490 / ± 125 / 1300
Hauptspindel / Drehzahl in [min^{-1}]	ISM 76 / 5000

Ein Foto dieser Maschine ist in der Abbildung 2-1 dargestellt.

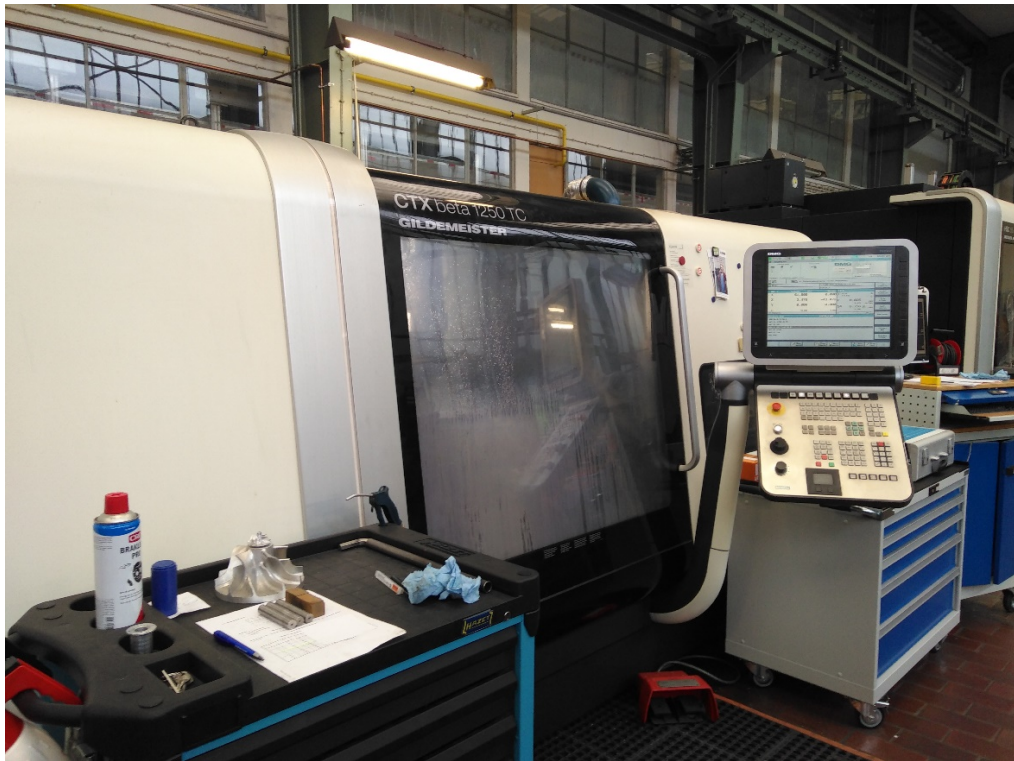


Abbildung 2-1: Foto der DMG MORI CTX Beta 1250 TC Siemens Maschine im IFQ Versuchsfeld OVGU

Als Rohling wurde ein Stab aus einer CoCrMo-Legierung mit dem Markennamen Ergiloy® 9.9135HL verwendet, der in der Prothetik verwendet wird. Alle Arten von Prothesen werden aus diesem Material hergestellt. Die chemische Zusammensetzung dieses Materials ist in Tabelle 2-2 angegeben:

Tabelle 2-2: Die chemische Zusammensetzung Ergiloy® 9.9135HL ISO 5832-12 (Masse in %) [48]

Material	Chemische Zusammensetzung								
Ergiloy® 9.9135HL	C	Si	Mn	Cr	Ni	Mo	Fe	N	Co
	0,14	1	1	26-30	1	5-7	0,75	0,25	bal

Die mechanischen Eigenschaften dieses Materials sind in Tabelle 2-3 angegeben:

Tabelle 2-3: Die mechanischen Eigenschaften Ergiloy® 9.9135HL ISO 5832-12 [49]

Material	Mechanische Eigenschaften	
Ergiloy® 9.9135HL	Härte in [GPa]	E-Modul in [GPa]
	4.9±0.2	235±10

Die Geometrie des bearbeiteten Stabes besteht aus drei Bereichen mit unterschiedlichen Durchmessern. Ein Abschnitt mit einem Durchmesser von $\varnothing 40$ mit einer Länge von 70 mm wurde als Schnittstelle zur Spannung des Rohlings in der Vorrichtung verwendet. Die Vorrichtung wurde in einem Nockenfutter fixiert und das Drehmoment über eine Längskeilverbindung auf das Werkstück übertragen. Für die Bearbeitung in diesem Experiment wurde ein Abschnitt von $\varnothing 65$ mit einer Länge von 25 mm verwendet. Ein Abschnitt von $\varnothing 60$ mit einer Länge von 45 mm wurde während früherer Experimente bearbeitet und in diesem nicht verwendet.

Abbildung 2-2 zeigt eine Skizze der Geometrie des Werkstücks.

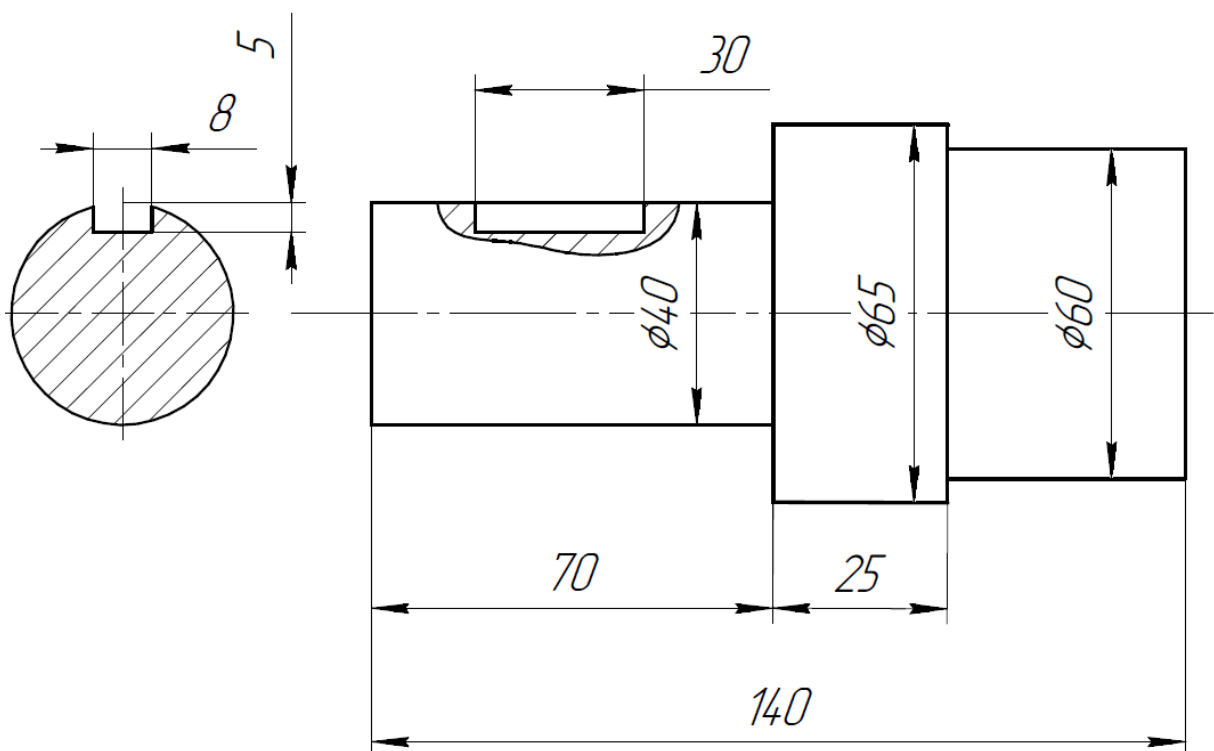


Abbildung 2-2: Werkstückskizze

Das Werkstück wurde mit einer Schraube in die Vorrichtung eingespannt. Die Berechnungen der Kräfte in dieser Verbindung wurden in früheren Studien durchgeführt.

In der Abbildung 2-3 ist die Vorrichtung geschildert:

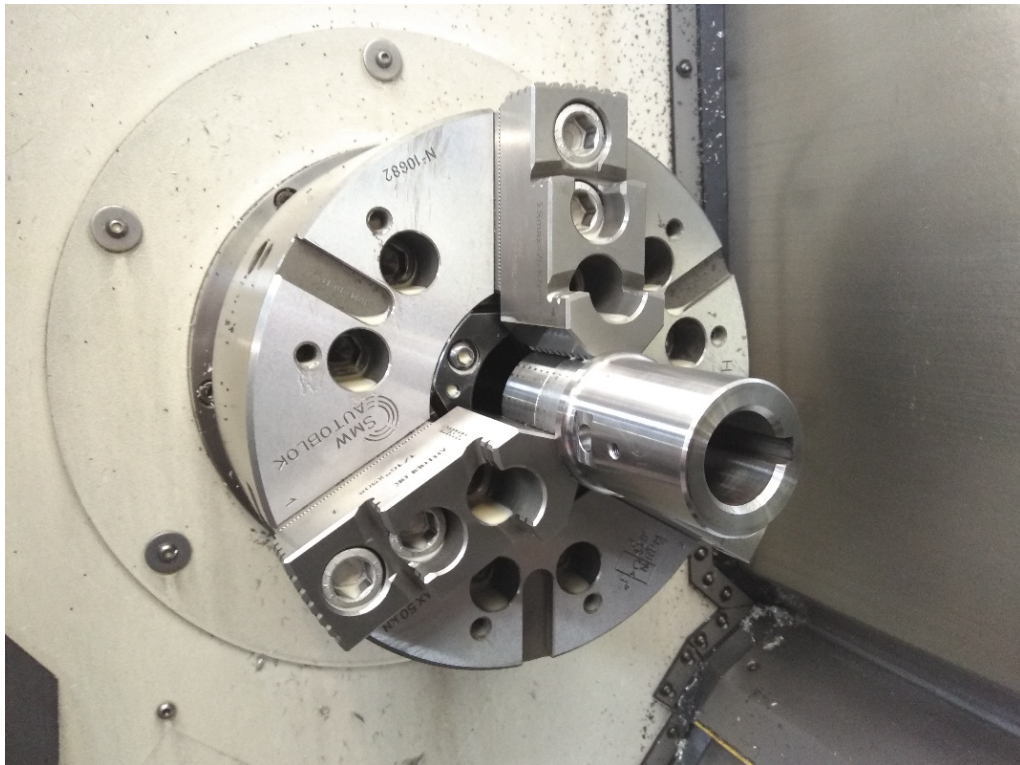


Abbildung 2-3: Vorrichtung

Abbildung 2-4 zeigt das in der Vorrichtung befestigte Werkstück.

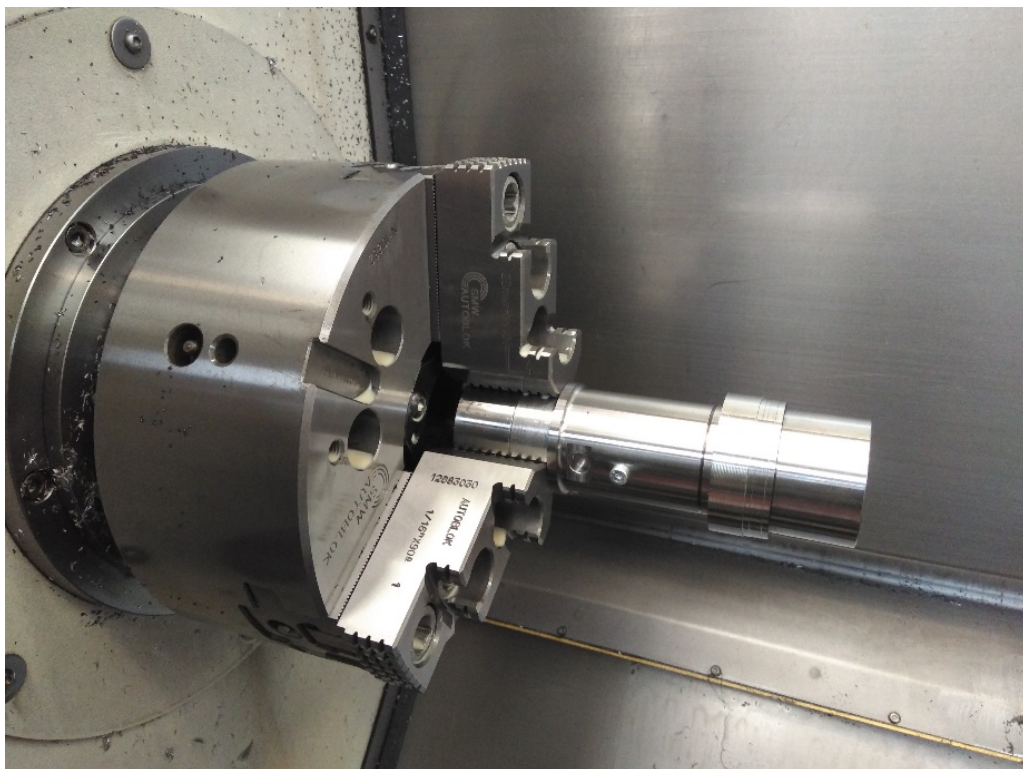


Abbildung 2-4: In der Vorrichtung befestigte Werkstück

Als Werkzeug wurde ein Sandvik CoroMill 390 Fräser mit einem Durchmesser von $\varnothing 10$ mm mit zwei Wendeschneidplatten P2.1.Z.AN, 175HB verwendet. Das Werkzeug wurde in einem intelligenten ProMicron-Spike-Werkzeughalter befestigt. Er ist mit Sensoren zur Messung von Schnittkraft, Biegemoment, Torsion und Temperatur ausgestattet. Über WLAN überträgt der Werkzeughalter ein Signal an einen Umwandler, der eine Verbindung zu einem Computer herstellt. Der vollständige Versuchsaufbau ist in Abbildung 2-5 dargestellt:

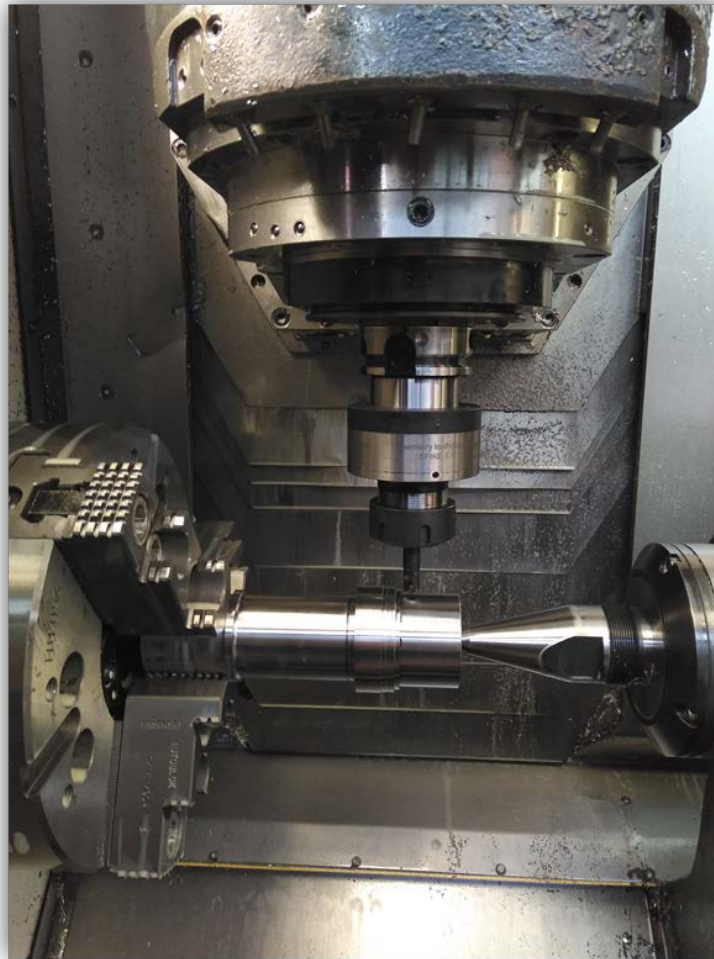


Abbildung 2-5: Versuchsaufbau

Das Werkstück wird während der Bearbeitung mit einer Pinole gedrückt.

2.3 Versuchsdurchführung

Wenn den Versuchsaufbau fertig war, hat die Versuchsdurchführung begonnen. Während der ersten beiden Experimente wurde die Bearbeitungslänge für nachfolgende Bearbeitungszyklen ausgewählt. Danach wurden 17 Versuche mit konstanter Bearbeitungslänge durchgeführt. Die Zerspanungswerte sind in Tabelle 2-4 aufgeführt:

Tabelle 2-4: Zerspanungswerte

e in [mm]	0
f in [mm]	0,07
n in [min ⁻¹]	9071

Die axiale Schnitttiefe nach jeder Bearbeitungsstufe variierte im Intervall $a_p = 0,05-0,47$.

Aus Gründen der Objektivität wurden die Wendeschneidplatten nach jedem Lauf ausgetauscht, unabhängig davon, ob sie verschlissen oder für die weitere Verarbeitung geeignet waren. Insgesamt wurden 20 Platten verwendet, da sie zweiseitig sind.

Nach jeder Bearbeitung wurde das System abgebaut, um die Verschleißtiefe des Werkzeugs und die Rauheit der behandelten Oberfläche zu messen.



Abbildung 2-6: Bearbeitungsprozess

2.4 Messung

Um die Verschleißtiefe des Instruments auf zwei Oberflächen zu messen, wurde ein digitales Mikroskop Keyence VHX 5000 verwendet. Der Messvorgang ist in den Abbildungen 2-7 und 2-8 zu sehen:



Abbildung 2-7: Messung der Verschleißtiefe (Draufsicht)



Abbildung 2-8: Messung der Verschleißtiefe (Oben)

Die Verschleißtiefe wurde danach auf den erzielten Bildern mithilfe des Keyence VHX 5000 Rechners gemessen. Das Beispiel für Messergebnisse wurde auf der Abbildung 2-9 dargestellt.



Abbildung 2-9: Verschleißmessung links: Oben; rechts: Draufsicht

Zur Messung der Oberflächenrauheit wurde ein NanoFocus-Gerät zur dreidimensionalen optischen Oberflächenmessung verwendet. Ein Foto des Messvorgangs ist in Abbildung 2-10 dargestellt.

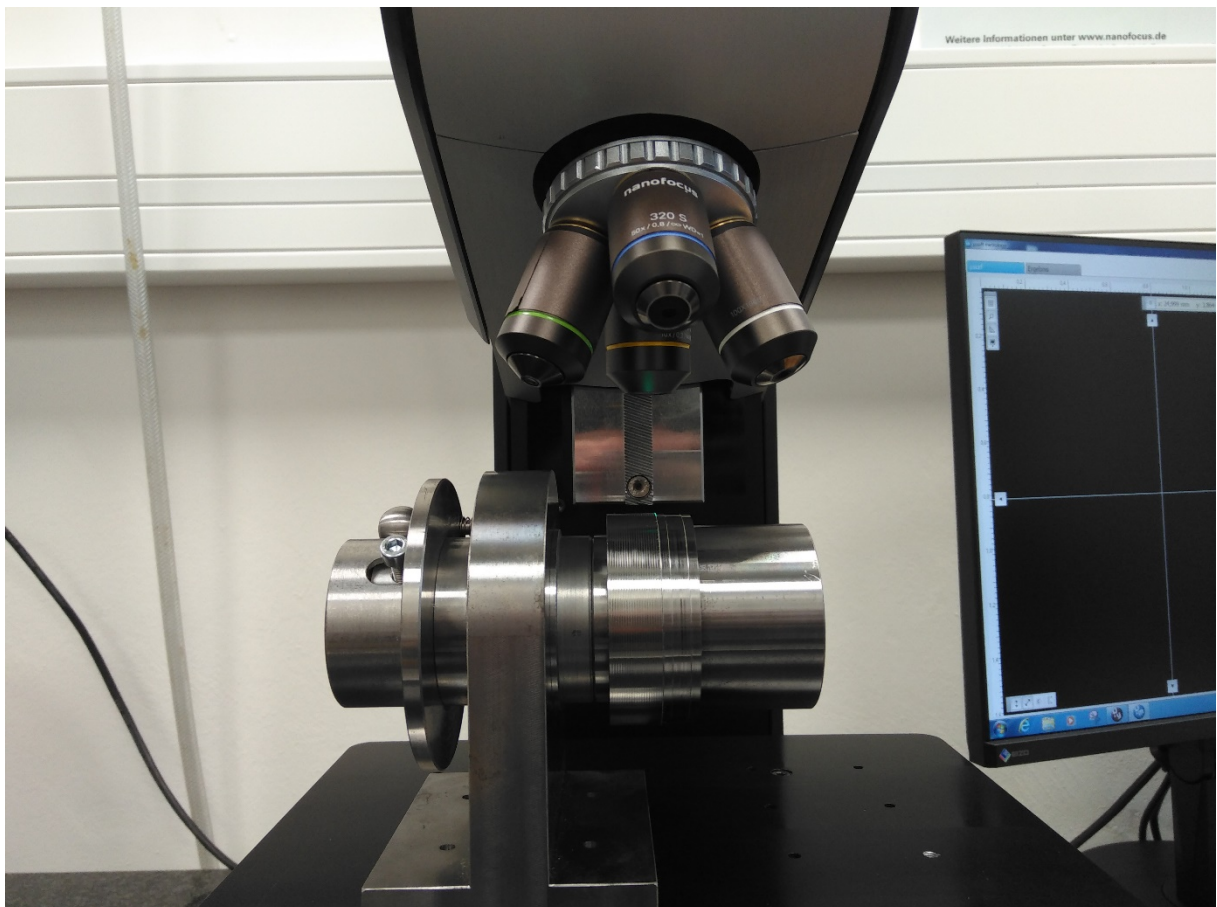


Abbildung 2-10: Messvorgangs von der optischen 3D-Rauheitsmessung mit NanoFocus

Mit der begleitenden Software wurde ein Messungsbericht pro Messung erstellt mit den allen Rauheitsparameter. Das Beispiel für Messergebnisse wurde auf der Abbildung 2-11 dargestellt.

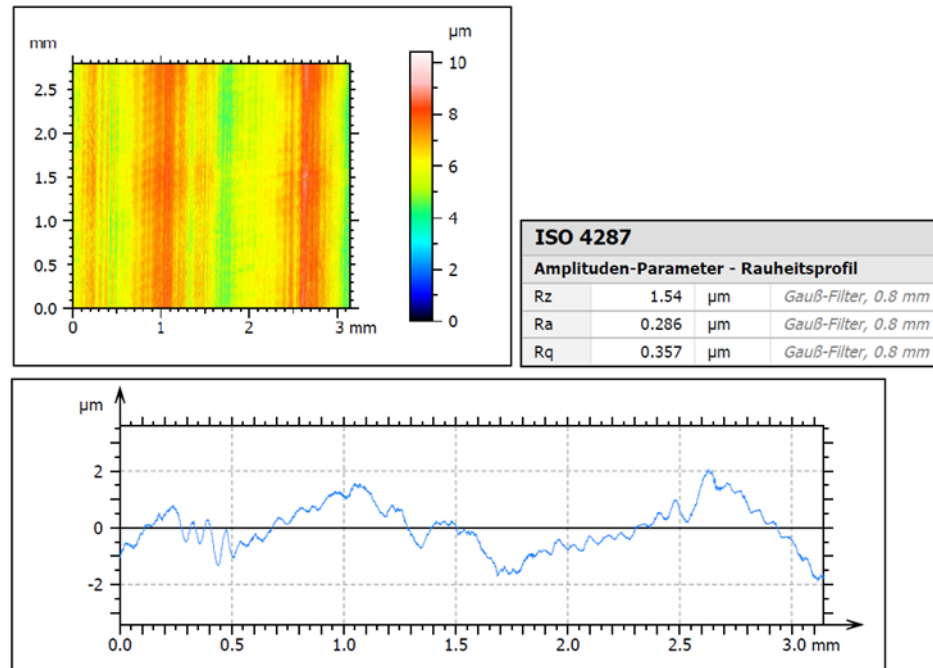


Abbildung 2-11: Optische 3D-Rauheitsmessung mit NanoFocus

In der Tabelle 2-5 sind die Ergebnisse aller Experimente angegeben:

Tabelle 2-5: Versuchsergebnisse

#	l in [mm]	ap in [mm]	Vo (WSP1) in [µm]	Vo (WSP2) in [µm]	Vd (WSP1) in [µm]	Vd (WSP2) in [µm]	Ra in [µm]
1	25	0,5	917	932	1018	1012	2,887
2	5	0,25	368	480	530	599	0,769
3	2,5	0,05	17	7	0	0	0,163
4	2,5	0,07	31	11	0	0	0,165
5	2,5	0,1	36	52	0	0	0,163
6	2,5	0,12	45	33	37	31	0,171
7	2,5	0,15	74	81	74	63	0,131
8	2,5	0,17	79	63	69	58	0,193
9	2,5	0,2	77	55	118	0	0,212
10	2,5	0,22	104	112	88	56	0,257
11	2,5	0,25	93	204	0	187	0,286

Tabelle 2-5: Versuchsergebnisse

#	l in [mm]	ap in [mm]	Vo (WSP1) in [µm]	Vo (WSP2) in [µm]	Vd (WSP1) in [µm]	Vd (WSP2) in [µm]	Ra in [µm]
12	2,5	0,27	104	203	122	73	0,211
13	2,5	0,3	127	125	92	75	0,177
14	2,5	0,32	132	154	113	70	0,289
15	2,5	0,35	109	105	159	122	0,312
16	2,5	0,37	98	146	117	169	0,297
17	2,5	0,4	0	56	0	108	0,167
18	2,5	0,45	166	257	115	211	0,272
19	2,5	0,47	183	201	106	202	0,321

Solcherweise wurden alle notwendigen Daten für festgestellte Zielfunktionen durch die Experimente gesammelt. Man kann bemerken, dass Verschleißtiefe (Oben) ist immer höher als Verschleißtiefe (Draufsicht). Das ist ein Grund für weitere Untersuchungen der Kraftwirkung, während der Bearbeitung.

Die nach den Experimenten gesammelten Daten sind Rohdaten und sie nicht geeignet und müssen weiterverarbeitet werden.

3 Analyse und Bearbeitung der Experimentenergebnisse

Dieser Schritt umfasst alle Aktivitäten zum Erstellen des endgültigen Datensatzes für Einsatz von Algorithmen.

Im Vergleich zu herkömmlichen Methoden, bei denen Extrahieren von Informationen aus Daten viel Zeit in Anspruch nimmt, wird in ML viel Zeit für die Aufbereitung der Daten aufgewendet, weil sie einen entscheidenden Einfluss auf die Ergebnisse haben [50]. Das ist der größte Teil des gesamten Analyseprojekts mit 50 bis 75 Prozent des Aufwands [16].

In diesem Teil wurde der Unterschied zwischen den Prozessen der Aufbereitung von Daten für Forschungszwecke und für industrielle Anwendungen am Beispiel der Vorverarbeitung der Ergebnisse des Drehfräsenexperiments untersucht. Dazu wurden alle Phasen der Datenvorverarbeitung in ML-Projekten berücksichtigt.

Ab diesem Moment beginnt die Verwendung der Programmierung in der Arbeit. Damit Forscher im Bereich Fertigungstechnik in ihren Projekten die ML-Techniken anwenden können, ohne tiefe Kenntnisse von Algorithmen und von der Programmiersprache, wurden Programme mit einer benutzerfreundlichen Schnittstelle entwickelt. Die Benutzer sollen in der Schnittstellebereich ihre Daten angeben und alle Komponenten in Algorithmen werden automatisch angepasst.

Auf solche Weise ist es möglich, interdisziplinäre ML-Forschungen ohne die Einbeziehung von Spezialisten aus verschiedenen Bereichen durchzuführen, was Personalkosten reduziert und den Zeitaufwand für die Kommunikation eliminiert.

3.1 Datenbewertung

Am Anfang sollen die Perspektiven für die Verwendung der gesammelten Daten zum Lernen von Algorithmen bewertet werden.

Es wurde 19 Versuche durchgeführt in dem experimentellen Teil der Arbeit. Diese kleine Datenmenge beschleunigt den Datenvorbereitungsprozess und führt schneller zum Training von Algorithmen. Es ermöglicht auch die Visualisierung und einfache Analyse der Funktion von Algorithmen und Ergebnisse.

Man kann dementsprechend den Korrelationsgrad von Werten zu bewerten, Fehler in der Versuchsplanung bzw. in dem Datenerfassungsprozess frühzeitig zu identifizieren und deren Verlauf rechtzeitig anzupassen. Beim Schreiben und Testen von Algorithmen ist es auch hilfreich zuerst mit kleinen Datensätze zuerst zu arbeiten.

3.2 Vorbereitung

3.2.1 Funktions- und Merkmalsauswahl

Zuerst sollen die Zielfunktionen identifiziert werden, die für Aufgabe relevant sind. Darüber hinaus wird in diesem Schritt auch Merkmalsauswahl durchgeführt. Die Merkmale, die nicht stark mit dem Ergebniswert korrelieren, die Genauigkeit des Modells manipulieren und beeinträchtigen. Deshalb sollen sie aus der Datensatz gelöscht werden.

Die Besonderheit der fertigungstechnischen Forschungen besteht darin, dass die Experimente im Voraus entsprechend der Forschungsaufgabe geplant sind und zielorientiert durchgeführt werden. Die Zielfunktionen und relevanten Merkmale wurden schon während der Versuchsplanung festgelegt. Das unterscheidet den Verlauf der Datenvorbereitung bei den Untersuchungen von dem Industrieanwendungsfall, wo die unterschiedlichen Daten im Laufe des Fertigungsablaufs ziellos erfasst werden und nur danach die Aufgabe gestellt wird, die interessierenden Kenntnisse aus diesen Daten zu erzielen. Deshalb ist hier im Gegensatz zu industriellen Prozessen keine zusätzliche Merkmalsauswahl erforderlich ist, was erheblich den Zeitaufwand in diesem Schritt vermindert.

Die entwickelten Algorithmen sollen nach dem Lernen aus den Daten Regressionsmodelle bilden und in der Lage sein, die Zielgröße gut vorherzusagen.

Da der vorliegende Datensatz kleine Versuchsanzahl enthält, werden fürs Lernen nur einparametrische Modelle gebildet. Diese Entscheidung erhöht die Wahrscheinlichkeit ein Muster in Daten zu erzielen und erleichtert die Visualisierung der Ergebnisse und dementsprechend ihre Interpretation, da nur maximal dreidimensionale Datensätze visualisiert werden können. Außerdem ermöglicht es die Funktionsweise einiger Algorithmen graphisch darzustellen und problemlos sie zu analysieren, was nur dann möglich ist, wenn Daten maximal zwei Parameter enthalten.

Manche ausgewählte Zielfunktionen haben entweder zwei Parameter oder mehr als eine Zielgröße und sollen daher in mehrere Funktionen zerlegt werden.

Es gibt aber Ausnahmefälle dabei. Für Verschleißtiefe wurden vier Werte erzeugt: Verschleiß auf zwei Flächen für zwei Wendeschneidplatten. Für die Reduzierung der Merkmalsanzahl wurden hier mehrere Merkmale zu einem zusammengefasst. Diese Methode erfordert die Fachkenntnisse. Darunter versteht man die Zusammenarbeit zwischen den Spezialisten im Bereich und Datenanalysten, sofern die Datenverarbeitung nicht direkt vom Spezialisten durchgeführt wird, wie es in diesem Projekt war.

Aus technischer Sicht ist es sinnvoll, den durchschnittlichen Verschleißwert jeder Oberfläche zu betrachten, weil diese Werte aufgrund der Prozessbedingungen (d.h. unterschiedliche Kraftwirkung) wesentlich unterscheiden.

Im untersuchenden Fertigungsprozess ist es sinnvoll auch Klassifikationsmodelle zu erzeugen nämlich Werkzeugzustandsklassifikation. Dazu wurden Wendeschneidplatte nach jedem Versuch als verschlissene oder nicht verschlissene manuell klassifiziert anhand der Verschleißtiefwerte.

3.2.2 Datenreinigung

Unter der Datenreinigung versteht man Suche und Behandlung der fehlenden, auffälligen oder redundanten Daten, weil sie negativ die Genauigkeit des Modells bewirken.

Im Forschungsprozess werden in der Regel alle Daten während der Datenerfassung in der Regel aufmerksam kontrolliert und deshalb gibt es kein Problem mit den fehlenden Daten in diesem Fall. Dies spart an dieser Stelle Zeit, die für die Auswahl der richtigen Behandlungsmethode mit fehlenden Daten aufgewendet werden könnte.

Während der Analyse bezüglich des Vorhandenseins redundanter Daten wurde herausgefunden, dass die ersten beiden Versuche unter anderen Bedingungen (andere Vorschubgeschwindigkeit und Bearbeitungslänge) durchgeführt wurden. D.h. diese Parameter in das Modell aufgenommen werden sollen, was unerwünscht ist. Deshalb wurden die Ergebnisse dieser Versuche aus den Daten entfernt.

Bei der Suche nach Ausreißer wurde bemerkt, dass die Ergebnisse des siebzehnten Versuchs auffällig aussehen. Deshalb wurde Entscheidung getroffen dieses Ergebnis für Regressionsmodelle zu löschen, damit es die Modellleistung nicht beeinflussen.

3.2.3 Datensatzerstellung

Für die ML-Algorithmen werden separate Datensätze aus den Rohdaten erstellt. Das wurde durch Programmierbefehle realisiert.

Nach dem Ausführen des Programmcodes werden die Datensätze in tabellarischer (Tabelle 3-1) und graphischer Form (Abbildung 3-1) dargestellt. Zur Vereinfachung der Datenanalyse sind alle Datensätze in aufsteigender Reihenfolge nach dem Merkmalswert sortiert.

Die Werte werden aus den Tabellen ausgenommen und als Variablen in der Form von Matrizen oder Vektoren im Programm gespeichert (die Merkmale unter Variabel X und Zielgröße unter Variabel Y).

Für die Klassifikationsalgorithmen sollen die Zielgröße bzw. Klasse als numerische Werte gespeichert werden, damit sie behandelt werden können (Tabelle 3-2). Da das Werkzeug zwei Hauptzustände (verschlissen oder nicht verschlissen) aufweisen kann, wurden die Klassen in der Binärform (0 oder 1) dargestellt (Abbildung 3-2).

Tabelle 3-1: Datensatz in tabellarischer Form (kontinuierliche Werte)

a_p in [mm]	Verschleiß (Oben) in μm	Verschleiß (Draufsicht) in μm
0,05	12	0
0,07	21	0
0,1	44	0
0,12	39	34
0,15	77,5	68,5
0,17	71	63,5
0,2	66	59
0,22	113	72
0,25	148,5	93,5
0,27	153,5	97,5
0,3	126	83,5
0,32	143	91,5
0,35	107	140,5
0,37	122	143
0,45	211,5	163
0,47	192	154

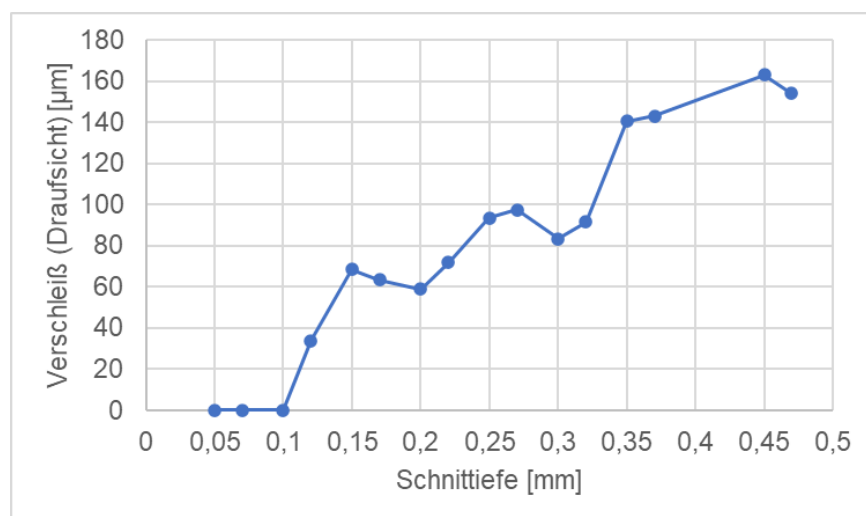


Abbildung 3-1: Graphische Darstellung des Datensatzes (kontinuierliche Werte)

Tabelle 3-2: Datensatz in tabellarischer Form (Klassen)

a_p in [mm]	Werkzeugzustand
0,05	0
0,07	0
0,1	0
0,12	0
0,15	0
0,17	0
0,2	0
0,22	0
0,25	1
0,27	1
0,3	1
0,32	1
0,35	1
0,37	1
0,4	0
0,45	1
0,47	1

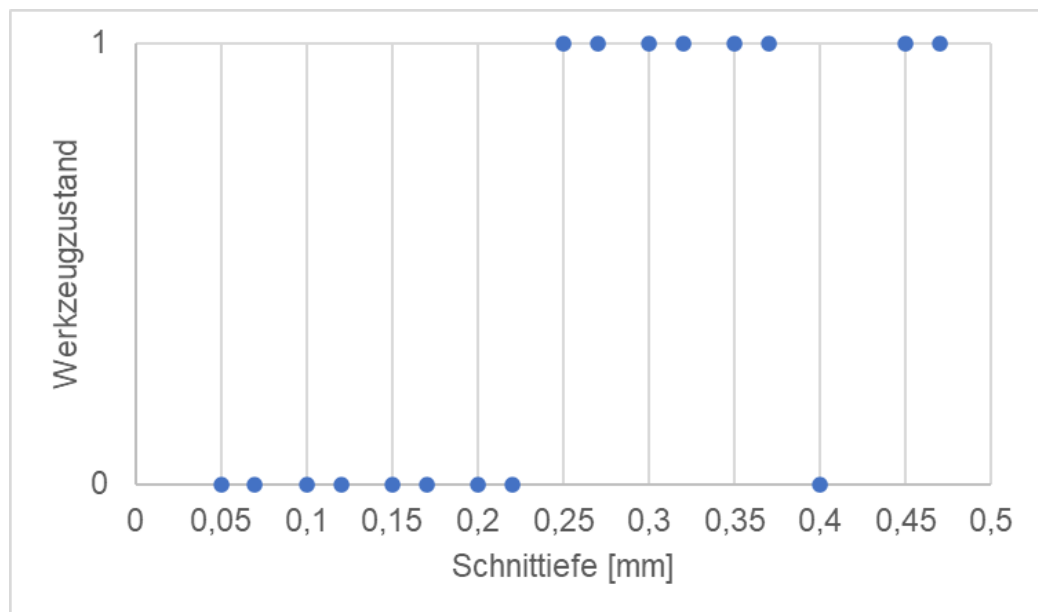


Abbildung 3-2: Graphische Darstellung des Datensatzes (diskrete Werte)

3.2.4 Codierung

Manche in dieser Arbeit betrachtete ML-Algorithmen können nur die numerischen Daten behandeln. Aus diesem Grund soll die Codierung durchgeführt werden. Darunter versteht man die Umwandlung aller kategorialen Daten im Datensatz in binäre Daten. Wenn die Anzahl der kategorialen Merkmalwerte gleich n ist, sieht die Codierungsergebnis wie folgt aus:

Kategorialer Merkmalwert 1 = [1, 0, ..., 0]

Kategorialer Merkmalwert 2 = [0, 1, ..., 0]

...

Kategorialer Merkmalwert n = [0, 0, ..., 1]

In vorhandenen Datensätzen gibt es keine kategorialen Werte, sodass dieser Schritt übersprungen werden kann.

3.2.5 Datenskalisierung

Der wesentliche Unterschied der Maßstäbe von angegebenen Merkmalwerte kann einen negativen Einfluss auf die Leistung und Ergebnis des Lernprozesses haben [51]. Dies liegt daran, dass Merkmale mit größeren Werten während des Trainings einen größeren Einfluss haben und ihre Gewichtungskoeffizienten innerhalb des Modells künstlich überbewertet werden. Das kann zu einer inkorrekten Bestimmung der Korrelation und dementsprechend zu falschen Modellergebnissen führen

Es gibt zwei übliche Methoden, um alle Merkmale auf dieselbe Skalierung zu bringen: Datennormalisierung und -standardisierung [51].

Bei der Datennormalisierung wird ein tatsächlicher Wertebereich, den ein numerisches Merkmal annehmen kann, in einen Standardwertebereich umgewandelt, der typischerweise im Intervall [0, 1] liegt.

Im Allgemeinen sieht die Normalisierungsformel folgendermaßen aus:

$$\bar{x}^{(j)} = \frac{x^{(j)} - \min^{(j)}}{\max^{(j)} - \min^{(j)}} \quad (3-1)$$

Standardisierung ist das Verfahren, bei dem die Merkmalswerte neu skaliert werden, sodass sie die Eigenschaften einer Standardnormalverteilung haben. Zuerst wird der Mittelwert subtrahiert, und dann wird durch die Standardabweichung dividiert, sodass die resultierende Verteilung eine Einheitsvarianz aufweist. Im Gegensatz zur Normalisierung bindet die Standardisierung die Werte nicht an einen bestimmten Bereich.

Standardbewertungen von Merkmalen werden nach Formel 3-2 berechnet:

$$\hat{x}^{(j)} = \frac{x^{(j)} - \mu^{(j)}}{\sigma^{(j)}} \quad (3-2)$$

Für die Auswahl des Skalierungsverfahren sollen normalerweise beide ausprobiert werden, um zu erfahren, welches für angegebene Aufgabe besser geeignet ist. Wenn es nicht möglich ist wegen der Größe des Datensatzes oder des Zeitmangels, gilt als Faustregel: Standardisierung wird für ein Merkmal bevorzugt, wenn es manchmal extrem hohe oder niedrige Werte oft haben kann und fürs Unüberwachtes Lernen. In allen anderen Fällen ist eine Normalisierung vorteilhaft [13].

Die vorhandenen Datensätze enthalten nur ein Merkmal. Deswegen ist Datenskalisierung nicht zweckmäßig und werden Modelle gut funktionieren, wenn sie von den ursprünglichen Daten trainiert werden. Das erleichtert die Interpretation der Ergebnisse und spart Zeit in diesem Schritt.

3.3 Korrelationsprüfung

Leistung des resultierenden Modells hängt direkt von der Datenqualität ab. Korrelationsprüfung dient zur Evaluierung der Datenqualität. Die Korrelation bedeutet die statistische Beziehung zwischen zwei Variablen.

Für ML-Algorithmen ist es wichtig Korrelation sowohl zwischen den Merkmalen, als auch zwischen Merkmalen und Zielgrößen zu prüfen.

Die Leistung einiger Algorithmen kann sich verschlechtern, wenn zwei oder mehr Variablen eng miteinander verbunden sind, was als Multikollinearität bezeichnet wird. Ein Beispiel ist die lineare Regression, bei der eine der beleidigenden korrelierten Variablen entfernt werden sollte, um die Modellleistung zu verbessern. Die Korrelation zwischen Eingabevariablen und Ausgabevariablen wird herausgefunden, um einen Einblick zu erhalten, welche Variablen (Merkmale) als Eingabe für die Entwicklung eines Modells relevant sein können oder nicht [52].

Zur Berechnung der Korrelationsstärke gelten Pearson- und Spearman-Korrelationskoeffizienten [52].

Der Pearson-Korrelationskoeffizient kann verwendet werden, um die Stärke der linearen Beziehung zwischen zwei Datenproben zusammenzufassen. Der Pearson-Korrelationskoeffizient wird als Kovarianz der beiden Variablen geteilt durch das Produkt der Standardabweichung jeder Datenprobe berechnet. Kovarianz wird als Durchschnitt des

Produkts zwischen den Werten jeder Probe berechnet, wobei die Werte zentriert wurden, d.h. deren Mittelwert subtrahiert wurde (Formel 3-3).

$$\text{Der Pearson-Korrelationskoeffizient} = \frac{\text{cov}(x; y)}{\text{stdev}(x) \times \text{stdev}(y)} \quad (3-3)$$

$$\text{cov}(x; y) = \frac{1}{n} \times \sum_{i=1}^n (x_i - \text{mean}(x)) \times (y_i - \text{mean}(y))$$

Der Spearman-Korrelationskoeffizient bewertet die monotone Beziehung zwischen zwei kontinuierlichen oder ordinalen Variablen. In einer monotonen Beziehung ändern sich die Variablen tendenziell zusammen, jedoch nicht unbedingt mit einer konstanten Geschwindigkeit. Er wird wie Pearson-Korrelationskoeffizient berechnet (Formel 3-4). Unterschied besteht darin, dass für die Bewertung der monotone Beziehung Daten im Voraus rangiert sein sollen.

$$\text{Der Spearman-Korrelationskoeffizient} = \frac{\text{cov}(\text{rg } x; \text{rg } y)}{\text{stdev}(\text{rg } x) \times \text{stdev}(\text{rg } y)} \quad (3-4)$$

$$\text{cov}(\text{rg } x; \text{rg } y) = \frac{1}{n} \times \sum_{i=1}^n (x_i - \text{mean}(\text{rg } x)) \times (y_i - \text{mean}(\text{rg } y))$$

Beide Koeffizienten geben einen Wert zwischen -1 und 1 zurück, der die Korrelationsgrenzen von einer vollständigen negativen Korrelation zu einer vollständigen positiven Korrelation darstellt. In der Tabelle 3-3 ist die Interpretation des Korrelationswerts angegeben.

Tabelle 3-3: Tabelle der Korrelationskoeffizienten und deren Interpretation [52]

Korrelationskoeffizient für direkte Beziehung	Korrelationskoeffizient für indirekte Beziehung	Stärke der linearen Beziehung
0.0	0.0	keine/triviale
0.1	-0.1	schwache/kleine
0.3	-0.3	mäßige/mittlere
0.5	-0.5	starke/große
1.0	-1.0	perfekte

Da die Pearson- und Spearman-Korrelationen die statistische Hypothesenteste sind, bei denen davon ausgegangen wird, dass keine Beziehung zwischen den Variablen besteht (Nullhypothese), wird dabei auch der p-Wert berechnet. In dieser Situation ist er wie folgt interpretiert [52]:

- $p\text{-Wert} \leq \alpha$: signifikantes Ergebnis, Nullhypothese ist abgelehnt, es gibt eine Beziehung (H1).
- $p\text{-Wert} > \alpha$: kein signifikantes Ergebnis, Nullhypothese kann nicht abgelehnt werden, es gibt keine Beziehung (H0).

In dieser Untersuchung wurden die Korrelationskoeffizienten zwischen allen erfassenden Daten berechnet und ihre Werte sind in dementsprechende Korrelationsmatrize eingetragen. Es wurde festgestellt, dass Signifikanzniveau α gleich 0.05 ist.

Die Korrelationsmatrizen für Pearson- (Tabelle 3-4) und Spearman-Korrelationen (Tabelle 3-5) werden mithilfe der Heatmap dargestellt. P-Werte sind in Klammern angegeben.

Tabelle 3-4: Pearson-Korrelationsmatrize

	a_p	Verschleiß (Oben)	Verschleiß (Draufsicht)	Ra
a_p	1	0.92 (0.001)	0.96 (0.0001)	0.73 (0.039)
Verschleiß (Oben)	0.92 (0.001)	1	0.88 (0.004)	0.64 (0.089)
Verschleiß (Draufsicht)	0.96 (0.0001)	0.88 (0.004)	1	0.75 (0.03)
Ra	0.73 (0.039)	0.64 (0.089)	0.75 (0.03)	1

Tabelle 3-5: Spearman-Korrelationsmatrize

	a_p	Verschleiß (Oben)	Verschleiß (Draufsicht)	Ra
a_p	1	0.88 (0.004)	0.95 (0.0004)	0.75 (0.038)
Verschleiß (Oben)	0.88 (0.004)	1	0.92 (0.001)	0.61 (0.1)
Verschleiß (Draufsicht)	0.95 (0.0004)	0.92 (0.001)	1	0.73 (0.04)
Ra	0.75 (0.038)	0.61 (0.1)	0.73 (0.04)	1

Aus den Korrelationstabellen werden folgende Schlussfolgerungen gemacht:

- 1) Es ist angenommen, dass keine linearen oder monotonen Beziehungen zwischen R_a und Verschleiß (Oben) existieren, weil p-Werte der Pearson- und Spearman-Korrelationen niedriger als Signifikanzniveau sind
- 2) Korrelationen sowohl zwischen R_a und a_p , als auch zwischen Verschleiß (Oben) und Verschleiß (Draufsicht) sind stark monotone, aber nicht lineare, weil dabei Spearman-Korrelationswerte höher als Pearson-Korrelationswerte sind.
- 3) Alle anderen Werten haben miteinander starke lineare Korrelation

3.4 Datenzerteilung

Da das Ziel des ML-Projekts ist einen Algorithmus und seine Hyperparameter so auszuwählen, dass er neue Beispiele am besten vorhersagt, ist es erforderlich für Leistungsevaluierung und Ergebnisvalidierung die Testdaten zu haben. Für die Erzeugung des Testdatensatzes sollen die gesammelten Daten geteilt werden. Es gibt dazu einige Methoden.

In den meisten praktischen Anwendungen in der Industrie wird „Train-Validation-Test Split“-Verfahren eingesetzt [53]. Dabei werden die verfügbaren Beispiele in drei Sätze aufgeteilt.

Bevor die Daten aufgeteilt werden, ist es wichtig, dass alle Zeilen im Dataset zufällig sortiert werden. Dies hilft, Verzerrungen im Modell zu vermeiden und objektives Bild beim Testen zu erzielen [12].

Der erste Satz (Trainingssatz) wird zum Trainieren unseres Algorithmus, der zweite (Validierungssatz) für die Auswahl des Algorithmus und der dritte (Testsatz) für die Auswahl Hyperparameterwerte verwendet. Der Trainingssatz soll der größte sein. Die Validierungs- und Testsätze sollen ungefähr gleich groß und viel kleiner als der Trainingssatz sein. Leistung des Ausgabeprädiktors für den dritten Satz gilt als Schätzer für den wahren Fehler des gelernten Prädiktors [13]. D.h. sobald das Modell die Werte für dritten Satz angemessen vorhersagen kann, ist es für die Implementierung in reale Umgebung (z.B. in die Produktion) bereit.

Der vorhandene Datensatz enthält nur sechzehn fürs Lernen gültige Beispiele und die Teilung in drei Sätze bewirkt durch Trainingsdatenmangel negativ die Lernleistung.

Eine effektive Alternative für geringe Datenmenge ist „K-fold cross-validation“-Verfahren, das in dieser Arbeit verwendet wurde. Zuerst werden die Werte der Hyperparameter festgestellt, die ausgewertet werden sollen. Anschließend soll man Datensatz in k Teile derselben Größe aufteilen. Jeder Teil wird als Falte (fold) bezeichnet [13].

Für acht Beispiele ist es sinnvoll sie in 4 Falten teilen $\{F1, F2, \dots, F4\}$. Jedes F_k , $k = 1, \dots, 4$ enthält 25% der Daten. Dann wurden vier Modelle wie folgt trainiert. Um das erste Modell f_1 zu trainieren, wurden alle Beispiele aus den Falten $F2, F3$ und $F4$ als Trainingssatz und die Beispiele aus $F1$ als Validierungssatz verwendet. Um das zweite Modell f_2 zu trainieren, wurden die Beispiele aus den Falten $F1, F3$ und $F4$ zum Trainieren und die Beispiele aus $F2$ als Validierungssatz verwendet. So wurden weiterhin iterativ Modelle wie diese erstellt und den Wert der interessierenden Metrik für jeden Validierungssatz von $F1$ bis $F4$ berechnet. Dann wurden die vier Werte der Metrik gemittelt, um den endgültigen Wert zu erhalten.

Beispiel der Datenteilung für 4-fold cross-validation ist in die Tabelle 3-6 eingetragen. Der Trainingssatz ist mit der blauen Farbe und Validierungssatz ist mit der orangen Farbe bezeichnet. Unter X-Variable ist axiale Schnitttiefe a_p in [mm] und unter Y-Variable ist Verschleiß (Oben) in [μm] angegeben.

Tabelle 3-6: Beispiel der Datenteilung beim „4-fold cross-validation“-Verfahren

Iteration 1		Iteration 2		Iteration 3		Iteration 4		Falte Nr.
X	Y	X	Y	X	Y	X	Y	
0,05	12	0,05	12	0,05	12	0,05	12	F1
0,07	21	0,07	21	0,07	21	0,07	21	
0,1	44	0,1	44	0,1	44	0,1	44	
0,12	39	0,12	39	0,12	39	0,12	39	
0,15	77,5	0,15	77,5	0,15	77,5	0,15	77,5	F2
0,17	71	0,17	71	0,17	71	0,17	71	
0,2	66	0,2	66	0,2	66	0,2	66	
0,22	113	0,22	113	0,22	113	0,22	113	
0,25	148,5	0,25	148,5	0,25	148,5	0,25	148,5	F3
0,27	153,5	0,27	153,5	0,27	153,5	0,27	153,5	
0,3	126	0,3	126	0,3	126	0,3	126	
0,32	143	0,32	143	0,32	143	0,32	143	
0,35	107	0,35	107	0,35	107	0,35	107	F4
0,37	122	0,37	122	0,37	122	0,37	122	
0,45	211,5	0,45	211,5	0,45	211,5	0,45	211,5	
0,47	192	0,47	192	0,47	192	0,47	192	

Alle Daten für jede Zielfunktion wurden auf ähnliche Weise wie in Tabelle 3-6 erstellt und tabellarisch aufgeführt. In diesem Teil wurden die Daten analysiert und für ihre weitere

Verwendung fürs Lernen der ML-Algorithmen bearbeitet. Hierzu wurden alle in den ML-Projekten verwendeten Maßnahmen durchgeführt.

4 Entwicklung und Konzipierung von ML-Systemen

In dieser Phase wurde die Entwicklung und Auswahl effektiver Algorithmen für die Datensätze durchgeführt, die für die weiteren Forschungen erforderlich sind. Das Kriterium für die Auswahl der Algorithmen waren die Ergebnisse der Validierung. Alle Algorithmen wurden mithilfe der Programmiersprache Python geschrieben.

Es wurde auch ein Vergleich der Genauigkeit von Algorithmen für maschinelles Lernen und herkömmlicher statistischer Methoden durchgeführt. Als statistische Methode wurde die Erstellung eines linearen Modells nach der Methode der kleinsten Quadrate gewählt.

Darüber hinaus, werden in diesem Teil die Konzepte der weiteren Verwendung der entwickelten Algorithmen vorgestellt, um die wissenschaftliche Erforschung des Drehfräsens zu optimieren und diesen Prozess in der Produktion zu steuern.

4.1 Vorwahl der Algorithmen nach Aufgabekriterien

Es ist nicht möglich, im Voraus vorherzusagen, welcher Algorithmus mit den ausgewählten Daten am besten funktioniert. Im Idealfall müssen alle Algorithmen ausprobiert und die Vorhersagegenauigkeit des Testdatensatzes verglichen werden [13]. Da die Anzahl der Algorithmen sehr groß ist und das Durchsuchen aller Optionen lange dauert, wurde eine Vorauswahl anhand der folgenden Kriterien durchgeführt, um den Prozess zu optimieren:

- Erklärbarkeit

Die genauesten Lernalgorithmen funktionieren wie sogenannte „Black Boxes“, da der Mechanismus, der die Eingabe in die Ausgabe umwandelt, durch eine figurative Box verschleiert wird [54]. Sie lernen Modelle, die nur sehr wenige Fehler machen, aber warum ein Modell eine bestimmte Vorhersage getroffen hat, kann sehr schwer zu verstehen und es gibt keine Möglichkeit die Modelle zu interpretieren, da innerhalb des Algorithmus mehrere komplizierten mathematischen Umwandlungen passiert. Beispiele für solche Modelle sind neuronale Netze oder SVM/SVR.

Diese Algorithmen sind nicht für die Forschungszwecke nutzbar, weil sie keine Informationen über die Beziehungen zwischen Daten und dementsprechend über den Prozess liefern. Sie sind aber für die industrielle Lösungen anwendbar, weil sie potentiell bessere Genauigkeit bei dem Vorhersagen erreichen können. In der Produktion ist egal, wie das Modell dabei aussieht. Ein Beispiel für solche Anwendungen sind Systeme für Adaptive Steuerung und Systeme für Echtzeitüberwachung des Werkzeugzustands.

Auf der anderen Seite erzeugen kNN-, lineare Regressions- oder Entscheidungsbaum-Lernalgorithmen Modelle, die nicht immer die genauesten sind, aber die Art und Weise, wie sie ihre Vorhersage treffen, ist einfach zu interpretieren.

Solche Algorithmen liefern die Informationen über die Beziehungen durch die Gewichtungskoeffizienten und nachvollziehbaren Lernprozess. Anhand dieser Kenntnisse ist es möglich die Modelle außerhalb dieser Algorithmen zu erstellen.

- Anzahl der Funktionen und Beispiele

Bei der Auswahl der Algorithmen soll man auch darauf achten, wie viele Trainingsbeispiele im Datensatz vorhanden sind und wie viele Merkmale jedes Beispiel charakterisiert. Einige Algorithmen, einschließlich neuronaler Netze und SVM, können eine große Anzahl von Beispielen und Millionen von Merkmalen verarbeiten. Andere können in ihrer Kapazität sehr beschränkt sein.

Vorhandene Datensätze enthalten wenige Trainingsbeispiele und Merkmale zu diesem Zeitpunkt. Es ist aber geplant die Modelle komplizierter zu machen und zukünftig sie für Produktion anzuwenden. Deshalb sollen die Algorithmen imstande sein, große Datenmengen zu verarbeiten.

- Nichtlinearität der Daten

Falls es eine Möglichkeit vorliegt zu prüfen, ob die Daten linear trennbar sind bzw. mit einem linearen Modell modelliert werden. Wenn ja, kann SVM mit dem linearen Kernel, der logistischen Regression oder der linearen Regression eine gute Wahl sein. Andernfalls funktionieren tiefe neuronale Netze möglicherweise besser für Daten.

In den erzeugten Datensätzen sind die Daten linear trennbar in der Klassifikationsmodellen und linear korreliert in der Regressionsmodellen. Deshalb ist es nicht zweckmäßig komplizierte nicht lineare Algorithmen, wie tiefe neuronale Netze, zu verwenden. Deshalb wurde für SVM mit dem linearen Kernel und für den logistischen Regression

In den erzeugten Datensätzen sind die Daten linear trennbar in der Klassifikationsmodellen und linear korreliert in der Regressionsmodellen. Deshalb ist es nicht zweckmäßig komplizierte nicht lineare Algorithmen, wie tiefe neuronale Netze, zu verwenden.

- Trainings- und Vorhersagengeschwindigkeit

Die Algorithmen von Maschinellen haben unterschiedliche Lerngeschwindigkeit. Es ist bekannt, dass neuronale Netze nur langsam trainieren. Einfache Algorithmen wie logistische und lineare Regression sowie das Lernen von Entscheidungsbäumen sind viel schneller.

In der Produktion sollen die Modelle die Vorhersagen schnell generieren, weil dort sehr hoher Durchsatz erforderlich ist. Einige Algorithmen wie SVMs, lineare und logistische Regression

oder einige Arten von neuronalen Netzen sind zum Vorhersagezeitpunkt extrem schnell. Einige andere, wie kNN, Ensemble-Algorithmen und sehr tiefe oder wiederkehrende neuronale Netze, können langsamer sein.

In dieser Arbeit entwickelten Algorithmen sollen in der Zukunft in der Produktion für Echtzeitüberwachung und adaptive Steuerung des Drehfräsenprozesses eingesetzt werden. D. h. die Algorithmen sollen die Vorhersagen schnell ausführen. Dementsprechend sind die Algorithmen wie kNN, Ensemble-Algorithmen und tiefe oder wiederkehrende neuronale Netze nicht für gestellte Aufgabe geeignet sind.

Basierend auf diesen Kriterien und den Zielen der Arbeit wurden 3 Regressionen (zur Vorhersage kontinuierlicher Werte) und 3 Klassifizierungsalgorithmen (zur Vorhersage diskreter Werte) ausgewählt und entwickelt.

4.2 Regressionsalgorithmen

4.2.1 Lineare Regression mithilfe des Konjugierte-Gradienten-Verfahrens

Lineare Regression mithilfe des Konjugierte-Gradienten-Verfahrens ist überwachte Lerntechnik, mit der die beste lineare Trendlinie zur Beschreibung eines Datensatzes gefunden wird [12].

Da sich im Stadium der vorläufigen Datenaufbereitung herausstellte, dass die gesammelten Daten hohe Werte des Pearson-Korrelationskoeffizienten aufweisen, wurde beschlossen, das lineare Modell auf Vorhersagen zu testen (Formel 4-1).

$$f_{w,b}(x) = w \times x + b \quad (4-1)$$

Das Modell wird verwendet, um das unbekannte y für ein gegebenes x wie folgt vorherzusagen: $y \leftarrow f_{w,b}(x)$. Die Modelle, die durch verschiedene Paare (w, b) parametrisiert sind, ergeben verschiedene Vorhersagen, wenn sie auf dasselbe Daten angewendet werden. Ziel ist die optimalen Werte (w, b) zu finden. Offensichtlich definieren die optimalen Parameterwerte das Modell, das die genauesten Vorhersagen macht. Ein geringerer Fehler zwischen dem tatsächlichen und dem vorhergesagten Wert anzeigt, dass der Algorithmus beim Lernen gute Arbeit geleistet hat [13].

Um diese Anforderung zu erfüllen, versucht das Optimierungsverfahren, mit dem die optimalen Werte für w und b gefunden werden, den mittleren quadratischen Fehler (MSE) zu minimieren (Formel 4-2).

$$MSE_{w,b} = \frac{1}{N} \times \sum_{i=1 \dots N} (f_{w,b}(x_i) - y_i)^2 \quad (4-2)$$

In einem kartesischen Koordinatensystem ist dies eine Gleichung für eine Parabel und kann grafisch dargestellt werden. Diese Funktion ist als Kostenfunktion bezeichnet, deren Wert von Parametern (Gewichtskoeffizienten) w und b abhängig ist.

Als Optimierungsverfahren wird dabei Konjugierte-Gradienten-Verfahren benutzt. Das ist ein iterativer Optimierungsalgorithmus, mit dem der Mindestwert für eine Funktion ermittelt wird. Die Idee besteht darin, dass durch die Berechnung der Ableitung / Steigung der Funktion das Minimum einer Funktion gefunden werden kann (Abbildung 4-1).

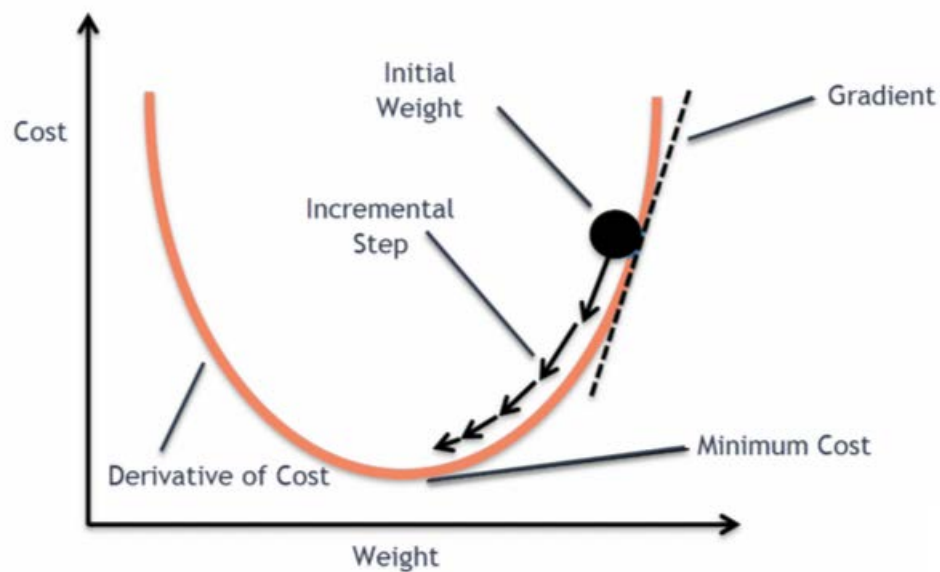


Abbildung 4-1: Graphische Darstellung des Optimierungsverfahrens bei dem Konjugierten-Gradienten-Verfahren [55]

Am Anfang werden die Gewichtskoeffizienten zufällig initialisiert. Während des Trainingsprozesses werden sich ihre Werte iterativ geringfügig ändern. Diese kleine Änderung sei mit δ bezeichnet.

Die Größe der Schritte (Größe der Parameterveränderung), die unternommen werden, um das Minimum oder den Boden zu erreichen, wird als Lernrate (η) bezeichnet. Mit größeren Schritten / höherer Lernrate mehr Bereich abdecken, laufen jedoch Gefahr, die Minima zu überschreiten. Andererseits benötigen kleine Schritte / kleinere Lernraten viel Zeit, um den niedrigsten Punkt zu erreichen.

Als nächstes soll die Richtung der Optimierung bestimmt werden. Die Ableitung einer Funktion liefert Informationen über die Richtung in der Optimierung vorgehen muss, um sie zu minimieren. Optimierungsalgorithmen wie das Konjugierte-Gradienten-Verfahren verwenden Ableitungen, um tatsächlich zu entscheiden, ob die Gewichte erhöht oder verringert werden

sollen, um eine Zielfunktion zu erhöhen oder zu verringern. Der Richtungsvektor nennt man Gradient und besteht aus partiellen Ableitungen (Formel 4-3).

$$\nabla \text{MSE}_{w,b} \equiv \left[\frac{\partial \text{MSE}}{\partial w}, \frac{\partial \text{MSE}}{\partial b} \right] \quad (4-3)$$

Daraus folgt, dass die Parameteränderung in jeder Iteration nach Formel 4-4 berechnet wird:

$$\begin{aligned} \Delta w &= \eta \times \frac{\partial \text{MSE}}{\partial w} \\ \Delta b &= \eta \times \frac{\partial \text{MSE}}{\partial b} \end{aligned} \quad (4-4)$$

Die partiellen Ableitungen sind dabei:

$$\begin{aligned} \frac{\partial \text{MSE}}{\partial w} &= \frac{1}{N} \times \sum_{i=1 \dots N} (f_{w,b}(x_i) - y_i) \times x_i \\ \frac{\partial \text{MSE}}{\partial b} &= \frac{1}{N} \times \sum_{i=1 \dots N} (f_{w,b}(x_i) - y_i) \end{aligned} \quad (4-5)$$

Die Aktualisierung der Parameter soll iterativ wiederholen, bislang die Funktion ein Minimum erreicht (Formel 4-6). Alle Parameter während der Iteration müssen gleichzeitig aktualisiert werden.

$$\begin{aligned} w &= w - \Delta w \\ b &= b - \Delta b \end{aligned} \quad (4-6)$$

Dieser Algorithmus wurde mit einer Programmiersprache implementiert. Die Eingabe in den Algorithmus ist ein Datensatz und Hyperparameterwerte. Die Hyperparameter dieses Algorithmus sind konventionell Lernrate und Anzahl der Iterationen. Die Auswahl der Anzahl der Iterationen, um das Minimum zu erreichen, nimmt Zeit in Anspruch und erschwert das Einrichten des Algorithmus. Deshalb wurde in dem entwickelten Programm zur Steigerung der Produktivität der Algorithmus so modifiziert, dass die Anzahl der Iterationen automatisch bestimmt wird. Der Algorithmus stoppt automatisch, wenn das Minimum der Kostenfunktion erreicht ist. Unten ist eine Stoppbedingung für den Algorithmus angegeben:

$$\text{MSE}_{w,b}(j) - \text{MSE}_{w,b}(j-1) \leq t \quad (4-7)$$

In diesem Formelausdruck bezeichnet j die Iterationsnummer des Algorithmus und t ist die Toleranz ($t = 0,000001 \dots 0,0001$).

Der Lernprozess wurde zu Diagnosezwecken anhand von zwei Diagrammen (Abbildung 4-2 und Abbildung 4-3) visualisiert.

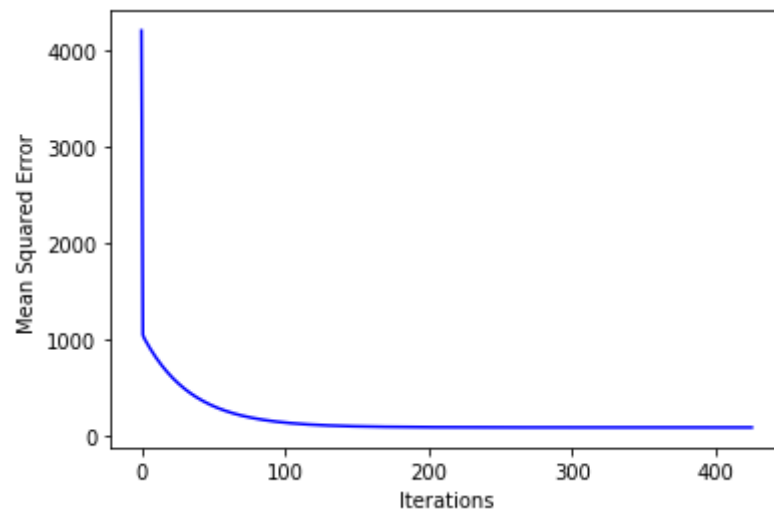


Abbildung 4-2: Die Veränderung von MSE im Laufe des Lernens

Abbildung 4-2 zeigt, wie sich der Wert von MCE während des Trainings ändert. Mit dieser Grafik kann man feststellen, ob der Lernprozess gut verläuft. Wenn der MCE-Wert mit jeder Iteration abnimmt, ist der Lernprozess erfolgreich. Andernfalls findet der Lernprozess nicht statt. Dies kann auf einen Fehler im Code oder einen großen Lernrate-Wert zurückzuführen sein.

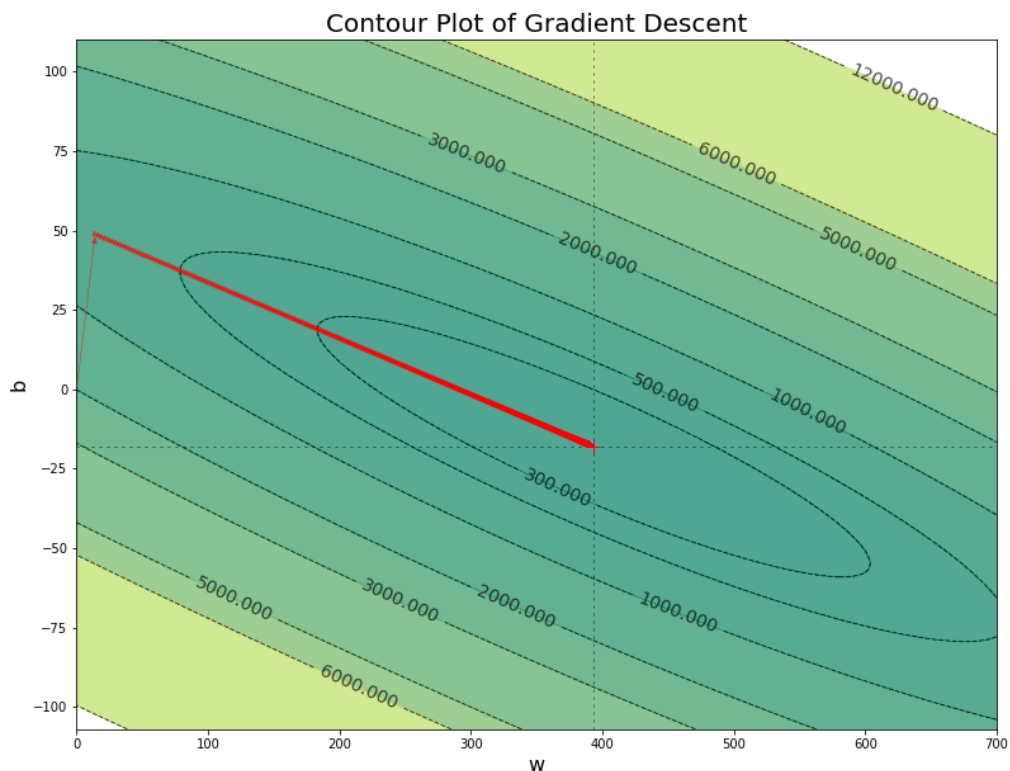


Abbildung 4-3: Konturdiagramm von Gradientenabstieg

Das Konturdiagramm des Gradientenabstiegs zeigt die Änderung der Werte der Gewichtskoeffizienten und der Position des Wertes der Funktion relativ zu ihrem Minimum nach dem Training. Auf diese Weise kann festgestellt werden, ob die Funktion ein Minimum erreicht hat. Der Aufbau dieses Diagramms ist nur bei Modellen mit einem Parameter möglich. In Modellen mit einer großen Anzahl von Parametern wird die Position des Werts der MCE-Funktion relativ zum Minimum durch Vergleichen dieser Werte geschätzt.

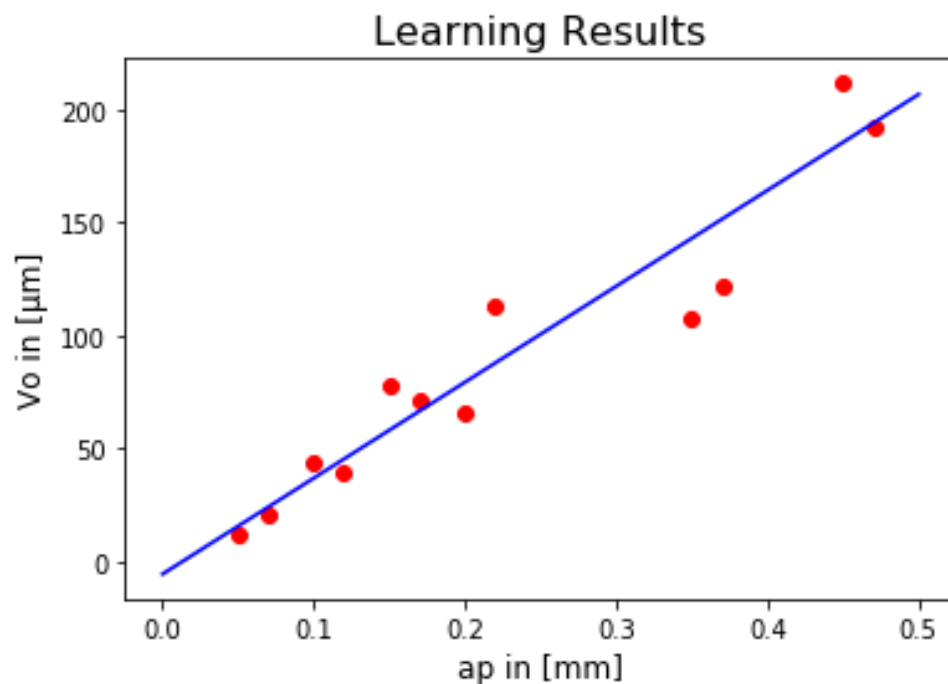


Abbildung 4-4: Beispiel des Trainingsergebnisses bei der Linealen Regression

Abbildung 4-4 zeigt das Ergebnis des Lernens des Algorithmus. Die als Ergebnis des Experiments erhaltenen Daten sind in der Grafik mit roten Punkten markiert. Die als Ergebnis des Trainings erhaltene Funktion wird durch eine blaue Linie angezeigt.

Nach dem Training wurde dieses Algorithmus durch das „4-fold cross-validation“-Verfahren für 4 Modelle durchgeführt. Die Wurzel aus dem mittleren quadratischen Fehler (RMSE) wurde als Metrik gewählt. Die Ergebnisse sind in Tabelle 4-1 angegeben.

Tabelle 4-1: Ergebnisse des „4-fold cross-validation“-Verfahrens in [µm]

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Gesamtwert
Vo(ap)	5,84	8,53	15,05	20,9	12,58
Vd(ap)	14,31	11,52	8,99	7,53	10,58
Ra(ap)	0,01	0,023	0,032	0,025	0,023
Ra(Vd)	0,049	0,024	0,026	0,017	0,029

Nach dem „4-fold cross-validation“ wurden alle Daten fürs Lernen benutzt und für alle Abhängigkeiten die folgenden Modelle erzielt (Tabelle 4-2):

Tabelle 4-2: Erzielte Modelle nach dem Lernen mittels Lineare Regression

	Modell
Vo(ap)	$V_o = 433,96 \times a_p - 1,29$
Vd(ap)	$V_d = 420,45 \times a_p - 21,23$
Ra(ap)	$Ra = 0,37 \times a_p + 0,13$
Ra(Vd)	$Ra = 0,001 \times V_d + 0,13$

4.2.2 Polynome Regression mithilfe des Konjugierte-Gradienten-Verfahrens

Bei der Modellerstellung mit Regression für eine einzelne Variable (bekannt als „Uni-Variate-Regression“) ist die Polynom-Regression eine wichtige Variante der Regressionsanalyse, die hauptsächlich über die lineare Regression als Leistungssteigerung dient [56]. Polynome Regression ermöglicht die Erstellung von nichtlinearen Modellen. Die Funktion, die der ML-Algorithmus lernt, ist eine Polynom bzw. Polynomgleichung (Formel 4-8).

$$f_{w,b}(x) = w_1 \times x + w_2 \times x^2 + w_3 \times x^3 + w_4 \times x^4 + \dots + w_n \times x^n + b \quad (4-8)$$

Das Polynom wird durch den Grad n bestimmt. Die lineare Regressionsfunktion ist ein Sonderfall des Polynoms, wenn $n = N-1$ ist. Gleichung aus Formel 4-8 kann in der Vektorform beschrieben werden (Formel 4-9):

$$f_{w,b}(x) = w^T \times x + b;$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \dots \\ x_n \end{bmatrix}; w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ \dots \\ w_n \end{bmatrix} \quad (4-9)$$

Für die Suche der optimalen Gewichtskoeffizienten wurde hier, wie bei der Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens eingesetzt. D.h. die Optimierung und Berechnungen nach Formeln 4-2 – 4-7 stattfinden.

Die Schwierigkeit besteht dabei in der Auswahl des Polynomgrades, bei dem Modell die neuen Daten am besten approximiert wird. Dazu muss der Algorithmus für alle Grade trainiert werden. Die maximale Komplexität einer Polynomfunktion ist der Grad von $n-1$, bei dem die Funktion alle Punkte im Trainingsdatensatz verbindet. Danach wird ein Grad der Polynomfunktion ausgewählt, bei dem maximale Genauigkeit erreicht wird. Das Auswahlproblem besteht darin, dass der nach dem Training erreichte MCE-Wert keine objektiven Informationen über die Genauigkeit dieses Modells für neue Daten liefert.

Der Trainingsalgorithmus macht die Kostenfunktion (d. H. Die Entfernung von jedem Punkt zur Linie) mit mehr Iterationen kleiner. Bei der Erhöhung der Funktionskomplexität er hat mehr Möglichkeiten dazu. Bei dem Polynomgrades $N-1$ kann die gelernte Funktion nach erfolgreichem Training alle Punkte verbinden und so einen Nullwert der Kostenfunktion erreichen. Dies bedeutet jedoch, dass die Linie in alle Punkte (einschließlich Rauschen) passt und sekundäre Muster erfasst, die für die Generalisierbarkeit des Modells nicht erforderlich sind. Das bewirkt negativ die Genauigkeit der Vorhersage neuer Daten.

Dieses Phänomen nennt man das Data-Overfitting. Overfitting bezieht sich auf ein Modell, das die Trainingsdaten zu gut modelliert. Das Data-Overfitting tritt auf, wenn ein Modell die Details und das Rauschen in den Trainingsdaten so weit lernt, dass sich das negativ auf die Leistung des Modells bei neuen Daten auswirkt [57]. Das bedeutet, dass das Rauschen oder zufällige Schwankungen in den Trainingsdaten vom Modell als Konzepte erfasst und gelernt werden. Das Problem ist, dass diese Konzepte nicht für neue Daten gelten und sich negativ auf die Fähigkeit des Modells zur Generalisierung auswirken. Overfitting ist bei nichtparametrischen und nichtlinearen Modellen wahrscheinlicher, die beim Erlernen einer Zielfunktion flexibler sind. Dazu gehört eigentlich die Polynome Regression.

Um dieses Phänomen zu identifizieren, wurde in der Arbeit eine Kreuzvalidierung verwendet, die sofort die Genauigkeit des Modells für neue Daten bestimmt. Mit den Lernkurven wird der optimale Grad des Polynoms (optimale Modellkomplexität) ausgewählt. Das ist die Grafik, die zeigt, wie sich die Größe der Kostenfunktion bei den Trainings- und Testdatensätzen ändert, wenn sich die Komplexität des Modells ändert.

Die optimale Komplexität des Modells ist eine, bei der der Unterschied zwischen den Werten der Kostenfunktion für die Trainings- und Testdatensätze minimal ist. Die Standardform der Lernkurven für die Polynomregression ist in der Abbildung 4-5 geschildert. Bis zu einem bestimmten Punkt nimmt der Wert der Kostenfunktion für beide Datensätze ab. Nach dem Überschreiten des optimalen Punkts wird der Kostenfunktionswert für den Trainingssatz niedriger, bis er Null erreicht, und für den Testdatensatz beginnt er zu wachsen.

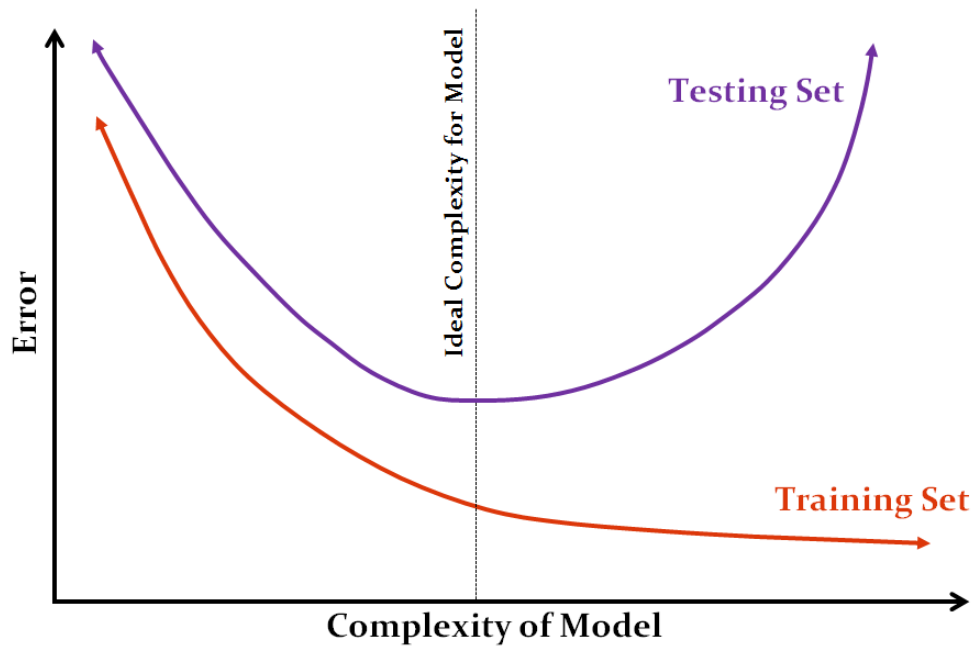


Abbildung 4-5: Lernkurven bei der Polynomen Regression [58]

Für jeden Grad des Polynoms wurden die Werte der Kostenfunktion bei jeder Iteration der Kreuzvalidierung bestimmt. Zur Erstellung der Lernkurven wurden Durchschnittswerte herangezogen.

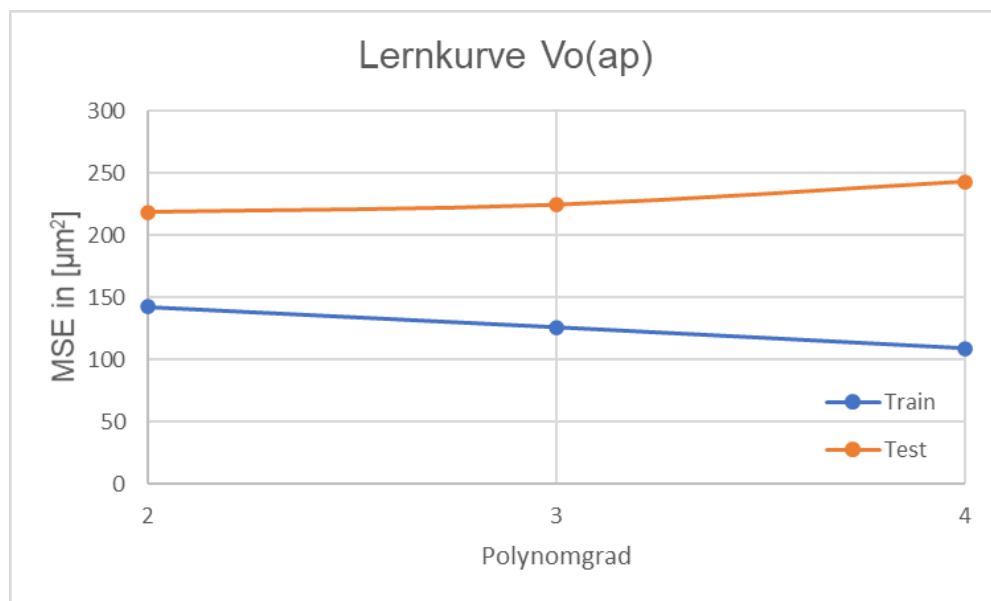


Abbildung 4-6: Lernkurven bei der Polynomen Regression für die Abhängigkeit V_o von a_p

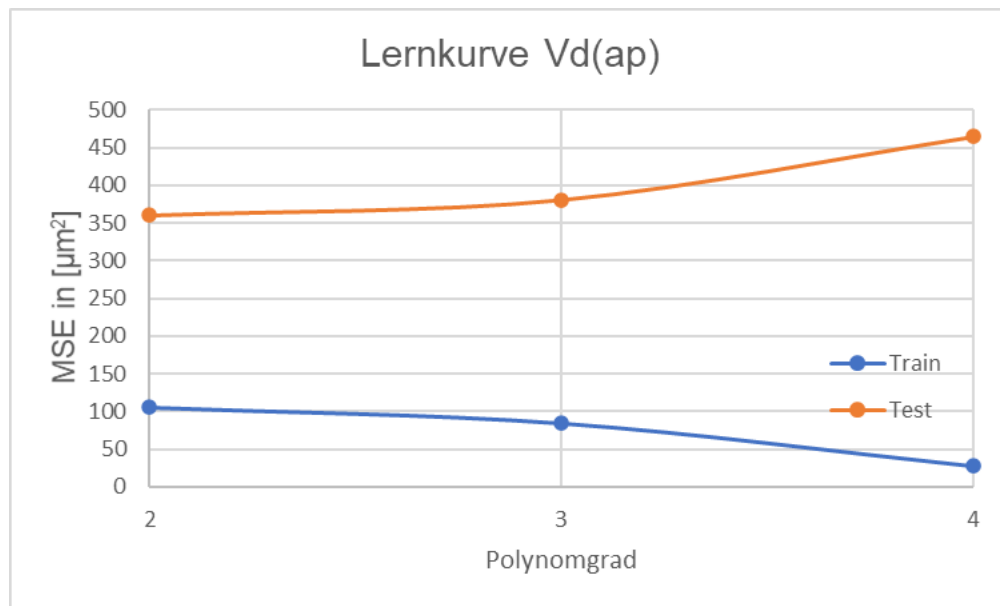


Abbildung 4-7: Lernkurven bei der Polynomen Regression für die Abhängigkeit V_d von a_p

Für die Abhängigkeiten der Verschleißtiefe auf beiden Oberflächen ist die optimale Komplexität des Modells bei dem Polynomgrad gleich zwei erreichbar. Der Grund dafür ist die starke lineare Korrelation zwischen diesen Größen (Abbildungen 4-6 und 4-7).

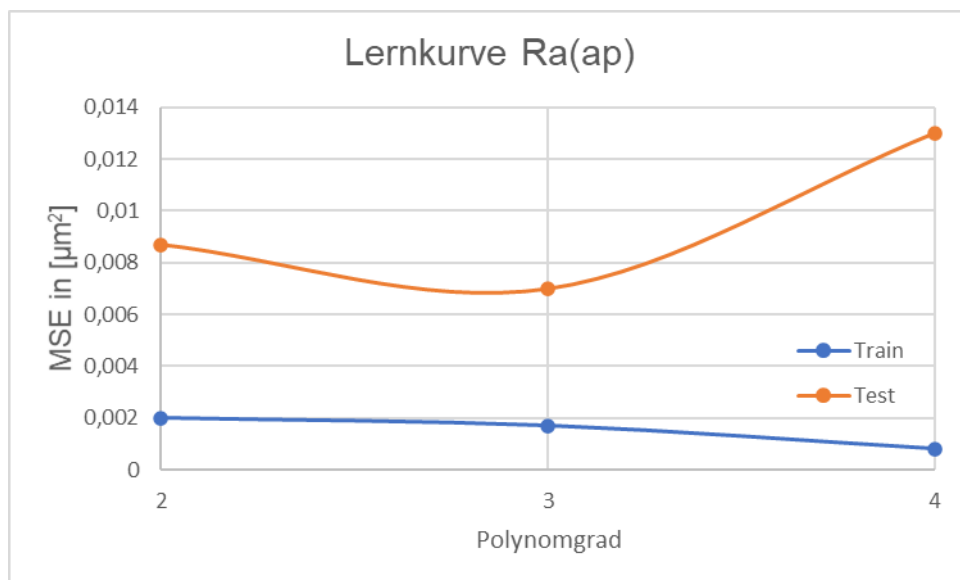


Abbildung 4-8: Lernkurven bei der Polynomen Regression für die Abhängigkeit R_a von a_p

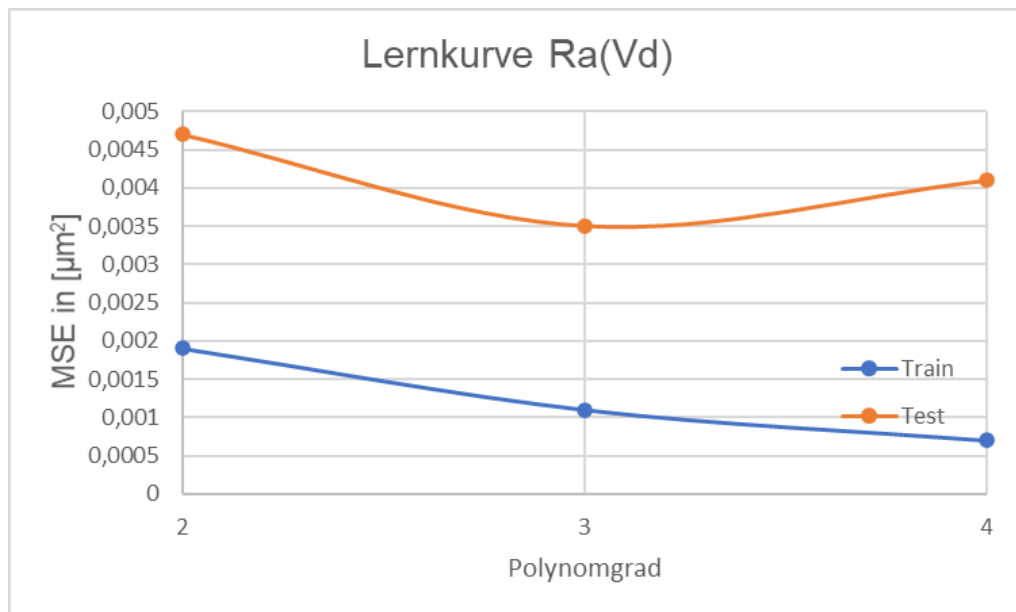


Abbildung 4-9: Lernkurven bei der Polynomen Regression für die Abhängigkeit Ra von Vd

Für die Abhängigkeiten der Oberflächenrauheit von der Schnitttiefe und von der Verschleißtiefe (Draufsicht) ist die optimale Komplexität des Models bei dem Polynomgrad gleich drei erreichbar (Abbildungen 4-8 und 4-9).

Nach der Auswahl von optimalen Polynomgraden wurden die Algorithmen gelernt. Für die Diagnose des Lernprozesses bei der Polynomen Regression kann nur ein Diagramm wegen der hohen Dimensionalität der Daten eingesetzt werden (Abbildung 4-10)

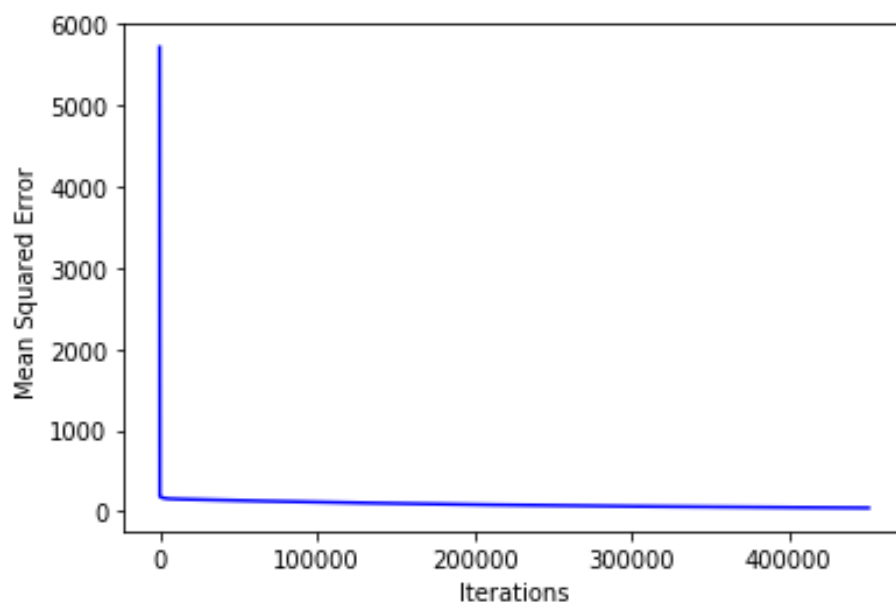


Abbildung 4-10: Die Veränderung von MSE im Laufe des Lernens

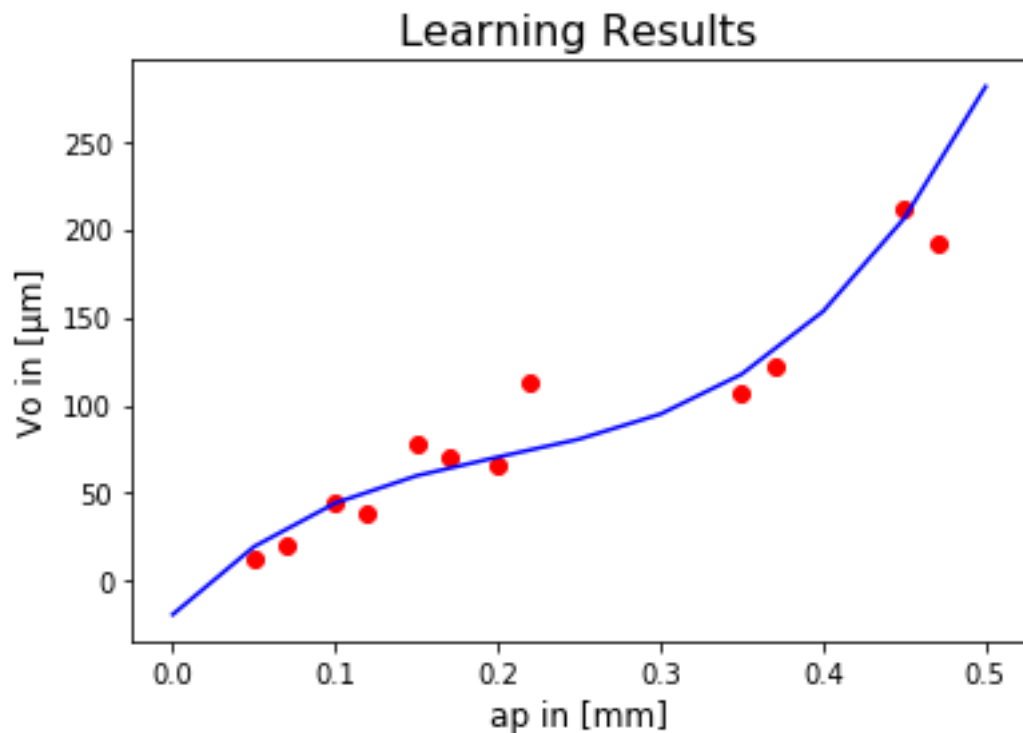


Abbildung 4-11: Beispiel des Trainingsergebnisses bei der Polynomen Regression

Abbildung 4-11 zeigt das Ergebnis des Lernens des Algorithmus bei der Polynomen Regression. Die als Ergebnis des Experiments erhaltenen Daten sind in der Grafik mit roten Punkten markiert. Die als Ergebnis des Trainings erhaltene Funktion wird durch eine blaue Linie angezeigt.

Nach dem Training wurde dieses Algorithmus durch das „4-fold cross-validation“-Verfahren für 4 Modelle durchgeführt. Die Wurzel aus dem mittleren quadratischen Fehler (RMSE) wurde als Metrik gewählt. Die Ergebnisse sind in Tabelle 4-3 angegeben.

Tabelle 4-3: Ergebnisse des „4-fold cross-validation“-Verfahrens in [μm]

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Gesamtwert
Vo(ap)	17,55	8,69	15,24	18,32	14,76
Vd(ap)	22,14	11,48	10,88	31,45	18,98
Ra(ap)	0,03	0,024	0,032	0,025	0,028
Ra(Vd)	0,08	0,038	0,026	0,09	0,059

Nach dem „4-fold cross-validation“ wurden alle Daten fürs Lernen benutzt und für alle Abhängigkeiten die folgenden Modelle erzielt (Tabelle 4-4):

Tabelle 4-4: Erzielte Modelle nach dem Lernen mittels Polynome Regression

	Modell
Vo(ap)	$V_o = 396,76 \times a_p + 74,99 \times a_p^2 + 2,13$
Vd(ap)	$V_d = 477,95 \times a_p - 117,44 \times a_p^2 - 26,42$
Ra(ap)	$R_a = 0,25 \times a_p + 0,11 \times a_p^2 + 0,04 \times a_p^3 + 0,15$
Ra(Vd)	$R_a = 1,28e-11 \times V_d + 1,06e-09 \times V_d^2 + 8,93e-8 \times V_d^3 + 2,34e-13$

4.2.3 Support Vector Regression (SVR)

Support Vector Regression (SVMs) basiert auf statistischem Lernen oder der Vapnik-Chervonenkis (VC) -Theorie und sind gut positioniert, um auf noch nicht gesehene Daten zu generalisieren. Einer der Hauptvorteile von SVR besteht darin, dass seine Rechenkomplexität nicht von der Dimensionalität der Eingabedaten abhängt. Darüber hinaus verfügt es über eine hervorragende Generalisierungsfähigkeit mit hoher Vorhersagegenauigkeit [59].

In den meisten linearen Regressionsmodellen besteht das Ziel darin, die Summe der quadratischen Fehler zu minimieren (z.B. Lineare Regression mithilfe des Konjugierte-Gradienten-Verfahrens). Im Gegensatz zu diesen Methoden besteht die Zielfunktion von SVR darin, die Koeffizienten zu minimieren, genauer gesagt die l2-Norm des Koeffizientenvektors. Zielfunktion ist dabei in der Formel 4-10 dargestellt.

$$\min \frac{1}{2} \times \|w\|^2 \quad (4-10)$$

Der Fehlerterm wird stattdessen in den Einschränkungen behandelt, in denen der absolute Fehler kleiner oder gleich einem bestimmten Rand festgelegt wird, der als maximaler Fehler ε (epsilon) bezeichnet wird (Formel 4-11).

$$\begin{aligned} y_i - w \times x_i - b &\leq \varepsilon \\ w \times x_i + b - y_i &\leq \varepsilon \end{aligned} \quad (4-11)$$

Man muss aber die Möglichkeit von Fehlern berücksichtigt werden, die größer als ε sind. Das wird mittels der Schlupfvariablen realisiert. Das Konzept der Schlupfvariablen ist folgendermaßen aussieht: Für jeden Wert, der außerhalb von ε liegt, kann man seine Abweichung vom Rand als ξ bezeichnen [59].

Es ist bekannt, dass diese Abweichungen das Potenzial haben zu existieren, ist es aber erwünscht sie dennoch so weit wie möglich zu minimieren. Somit werden diese Abweichungen zur Zielfunktion hinzugefügt (Formel 4-12).

$$\min \frac{1}{2} \times \|w\|^2 + C \times \sum_{i=1}^N |\xi_i| \quad (4-12)$$

Der Fehlerterm wird dementsprechend in den folgenden Einschränkungen behandelt (Formel 4-13):

$$\begin{aligned} y_i - w \times x_i - b &\leq \varepsilon + |\xi_i| \\ w \times x_i + b - y_i &\leq \varepsilon + |\xi_i| \end{aligned} \quad (4-13)$$

Die graphische Darstellung dieser Einschränkungen und Abweichungen ist in der Abbildung 4-12 geschildert.

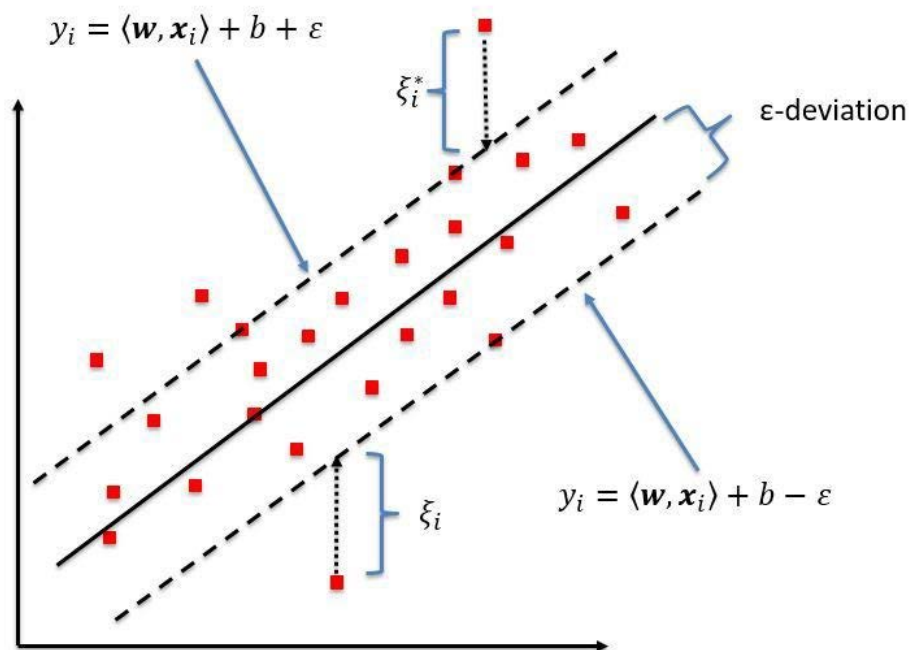


Abbildung 4-12: Darstellung der Ränder und Abweichungen bei der Support Vector Regression [60]

In der Zielfunktion gibt es zusätzlichen Hyperparameter, C, den eingestellt werden kann. Mit zunehmendem C steigt auch die Toleranz für Punkte außerhalb von ε . Wenn sich C zum Nullwert nähert, nähert sich die Toleranz zu der Null und die Gleichung kollabiert in die vereinfachte (wenn auch manchmal nicht realisierbare).

Dieser Algorithmus wurde in der Entwicklungsumgebung realisiert. Die Eingabe in den Algorithmus ist ein Datensatz und Hyperparameterwerte. Die Hyperparameter dieses

Algorithmus sind die Toleranz ϵ und Hyperparameter C für die Einstellung der Toleranz für Punkte außerhalb von ϵ .

Support Vector Regression Algorithmus gehört zu den sogenannten „Black Box“-Algorithmen [54]. Das bedeutet, dass es unmöglich ist den Lernprozess zu überwachen. Deshalb ist es möglich die Leistung des gelernten Models nur anhand der Ergebnisse zu evaluieren.

Das Ergebnis des Lernens des Algorithmus bei der Support Vector Regression ist in der Abbildung 4-13 angezeigt. Die als Ergebnis des Experiments erhaltenen Daten sind in der Grafik mit roten Punkten markiert. Die als Ergebnis des Trainings erhaltene Funktion wird durch eine blaue Linie angezeigt.

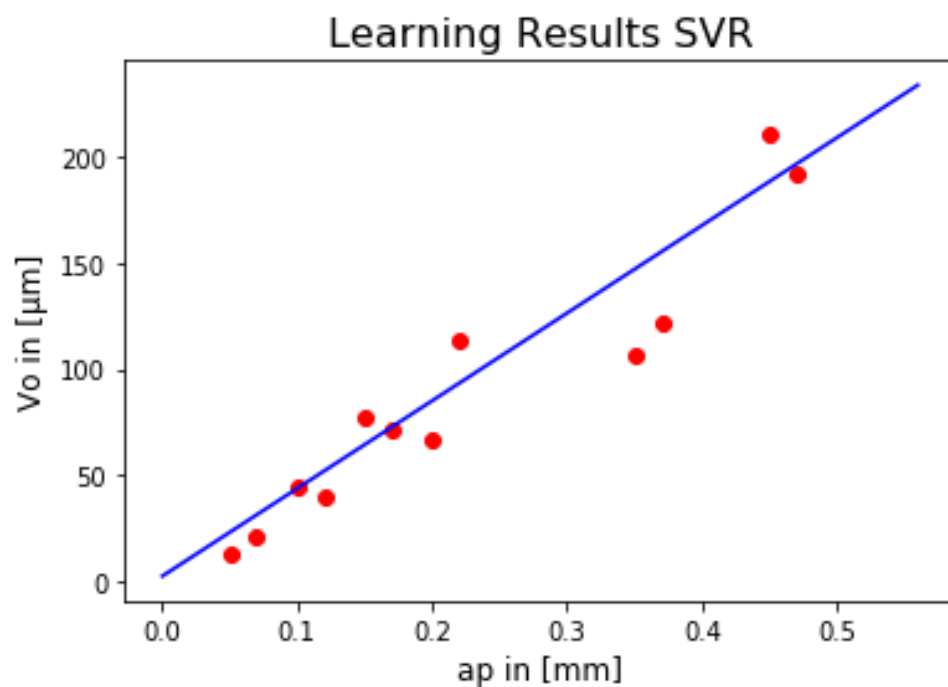


Abbildung 4-13: Beispiel des Trainingsergebnisses bei der Support Vector Regression

Nach dem Lernen wurde dieses Algorithmus durch das „4-fold cross-validation“-Verfahren für 4 Modelle validiert. Die Wurzel aus dem mittleren quadratischen Fehler (RMSE) wurde als Metrik gewählt. Die Ergebnisse sind in Tabelle 4-5 angegeben.

Tabelle 4-5: Ergebnisse des „4-fold cross-validation“-Verfahrens in [µm]

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Gesamtwert
Vo(ap)	8,4	8,79	14,04	17	12,06
Vd(ap)	20,92	10,92	8,67	14,23	13,68
Ra(ap)	0,004	0,02	0,031	0,052	0,027
Ra(Vd)	0,001	0,03	0,026	0,015	0,018

Die „Black-Box“-Funktionsweise des Algorithmus bedeutet, dass es nicht möglich die gelernten Modelle aus dem Algorithmus zu extrahieren. Die Vorhersagen werden nur mithilfe des gelernten Algorithmus erzielt. D.h. SVR liefert keine Informationen über die Beziehungen zwischen den Daten und somit ist nicht für theoretisch-orientierte Zwecke geeignet.

4.3 Statistisches Verfahren für Regressionsaufgabe

Für den Vergleich der ML- Algorithmen und konventionellen Statistik, wurde in dieser Arbeit eine lineare Funktion unter Verwendung der Methode der kleinsten Quadrate (OLS) erhalten. Die gewöhnliche Methode des kleinsten Quadrats sieht einfach aus und die Berechnung ist auch Leistungsfähig.

Diese Lösung ist jedoch nicht skalierbar. Die OLS-Methode wird so nur für univariate Datensätze funktionieren, bei denen es sich um einzelne unabhängige Variablen und einzelne abhängige Variablen handelt. Für andere Fällen werden die Formeln anders aussehen und sollen immer manuell bezogen werden. Die mangelnde Flexibilität macht diese Methode für praktische Aufgaben im Zusammenhang mit adaptiver Steuerung, Online-Qualitätskontrolle oder Zustandskontrolle ungeeignet. Bei theoretisch orientierten Problemen erschwert diese Methode die Verarbeitung hochdimensionale Daten erheblich.

Die Methode der kleinsten Quadrate ist das mathematische Standardverfahren zur Ausgleichsrechnung [61]. Das Ziel besteht darin, die Parameter einer Modellfunktion so anzupassen, dass sie am besten zu einem Datensatz passen. Die Modellfunktion ist in der Formel 4-1 dargestellt.

Die Koeffizienten w und b der bestangepassten Geraden sind gesucht bzw. die Koeffizienten mit der kleinsten Summe der Anpassungsfehlerquadrate (Quadrat der Residuen). Die Residuen sind die Abweichungen e_i zwischen der gesuchten Geraden und den jeweiligen Messwerten (Formel 4-14).

$$\min \sum_{i=1}^N e_i^2 \quad (4-14)$$
$$e_i = w \times x_i + b - y_i$$

Die Punktwolke, Residuen und Quadrate der Residuen werden in der Abbildung 4-14 dargestellt.

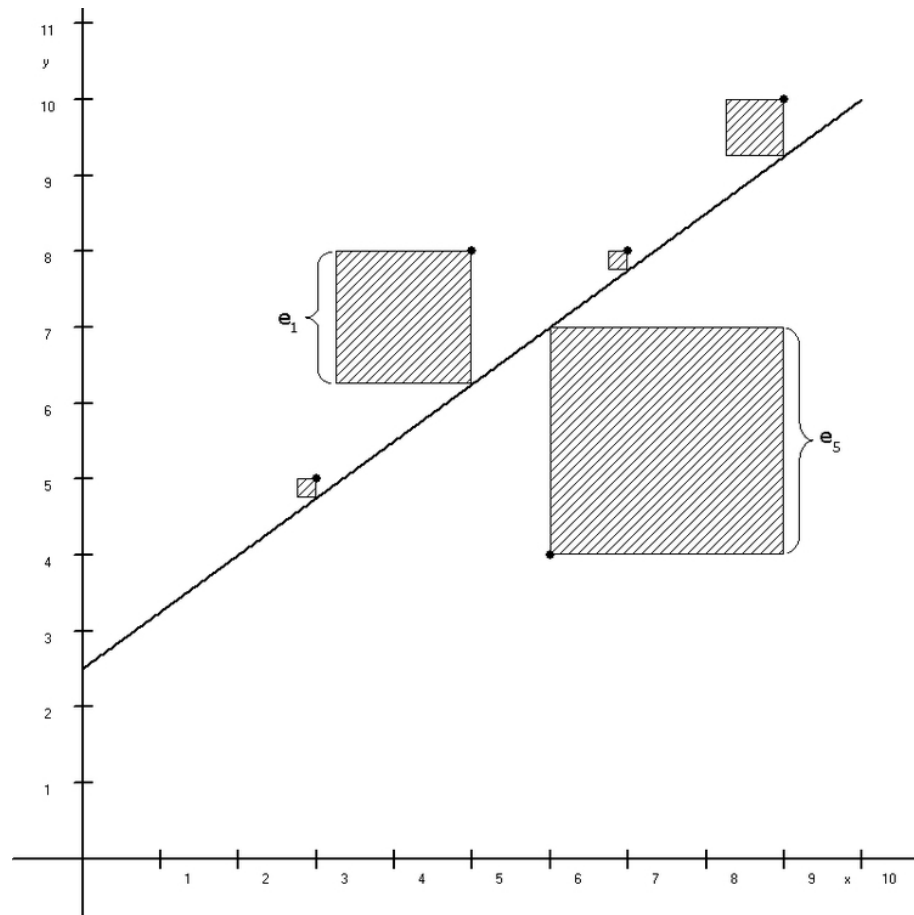


Abbildung 4-14: Punktwolke, Residuen und Quadrate der Residuen bei der Methode der kleinsten Quadrate

Im Gegensatz zu ML-Algorithmen werden die Parameterwerte w und b bei der Methode der kleinsten Quadrate nicht iterativ gefunden, sondern analytisch anhand von Formeln berechnet. Um den Parameter w zu finden, wird die Formel 4-15 verwendet.

$$w = \frac{SP_{xy}}{SQ_x} \quad (4-15)$$

Wobei SP_{xy} ist die Summe der Abweichungsprodukte zwischen x und y , die nach Formel 4-16 berechnet:

$$SP_{xy} = \sum_{i=1}^N (x_i - \bar{x}) \times (y_i - \bar{y}) \quad (4-16)$$

Die SQ_x ist die Summe der Abweichungsquadrate von x und sie wird nach Formel 4-17 berechnet:

$$SQ_x = \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4-17)$$

Nach der Berechnung des Parameters w kann man den anderen Parameter b nach Formel 4-18 finden:

$$b = \bar{y} - w \times \bar{x} \quad (4-18)$$

Bei den ML-Algorithmen wird also die numerische Approximation dieser Lösung durch iterative Methode gefunden, die der OLS-Lösung nahe (aber nicht genau gleich) wäre. Die Methode der kleinsten Quadrate berechnet die genaue Lösung.

Diese Berechnungen werden mithilfe der Programmiersprache realisiert werden. Die Eingabe des Programms ist dabei nur ein Datensatz. Es gibt keine Hyperparameter, die eingestellt werden sollen. Da dabei kein wirklicher Lernprozess stattfindet, kann man nur die Ausgabe des Programms zu evaluieren (Abbildung 4-15).

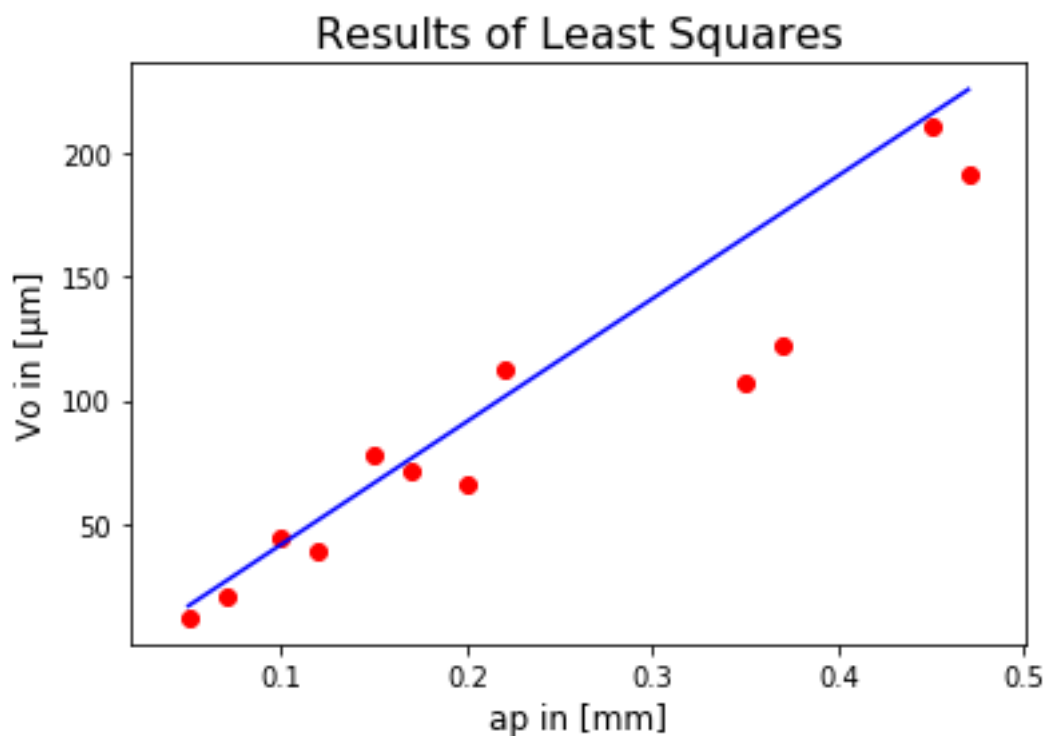


Abbildung 4-15: Beispiel des Berechnungsergebnisses bei der Methode der kleinsten Quadrate

Die Validierung der analytisch erzeugten Modelle wurde auch mittels des „4-fold cross-validation“-Verfahrens durchgeführt. Unterschied besteht nur darin, dass das größte Teil des Datensatzes nicht fürs Lernen, sondern für die analytische Berechnung. Die Wurzel aus dem mittleren quadratischen Fehler (RMSE) wurde als Metrik gewählt. Die Ergebnisse sind in Tabelle 4-6 angegeben.

Tabelle 4-6: Ergebnisse des „4-fold cross-validation“-Verfahrens in [μm]

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Gesamtwert
Vo(ap)	5,84	8,54	17,04	21	13,1
Vd(ap)	14,31	11,54	9	7,53	10,59
Ra(ap)	0,011	0,023	0,032	0,025	0,023
Ra(Vd)	0,027	0,031	0,026	0,023	0,027

Nach der Validierung wurden die Koeffizienten basierend auf einem vollständigen Datensatz berechnet. Die resultierenden analytischen Modelle sind in Tabelle 4-7 aufgeführt.

Tabelle 4-7: Erzielte Modelle nach dem Lernen mittels der Methode der kleinsten Quadrate

	Modell
Vo(ap)	$V_o = 433,95 \times a_p - 1,29$
Vd(ap)	$V_d = 420,45 \times a_p - 21,229$
Ra(ap)	$Ra = 0,368 \times a_p + 0,129$
Ra(Vd)	$Ra = 0,0009 \times V_d + 0,149$

4.4 Auswahl der Regressionsalgorithmen

Nachdem alle Algorithmen validiert worden waren, wurde eine Analyse der Ergebnisse durchgeführt und die endgültige Auswahl der Algorithmen getroffen, die in der Zukunft für praktische Anwendungen und theoretische Studien des Drehfräsens verwendet werden.

Da es keine universelle Lösung für alle Datensätze gibt, wird die Auswahl des optimalen Algorithmus für jede der vier untersuchten Abhängigkeiten separat getroffen. Gleichzeitig wird die Genauigkeit der Modelle verglichen, die unter Verwendung von Machine Learning Algorithmen erhalten und unter Verwendung der Methode der kleinsten Quadrate analytisch erhalten wurden.

Zur Verdeutlichung wurden alle Ergebnisse für jede Abhängigkeit in separaten Diagrammen zusammengefasst. Die Genauigkeit wird durch den Wurzelwert aus dem mittleren quadratischen Fehler gemessen. Je niedriger der RMSE-Wert ist, desto genauer ist das Modell. Manchmal ist der Unterschied zwischen den RMSE-Werten für verschiedene

Algorithmen so gering, dass es im Diagramm schwer zu erkennen ist. Daher werden die genauen Werte am unteren Rand des Diagramms angezeigt.

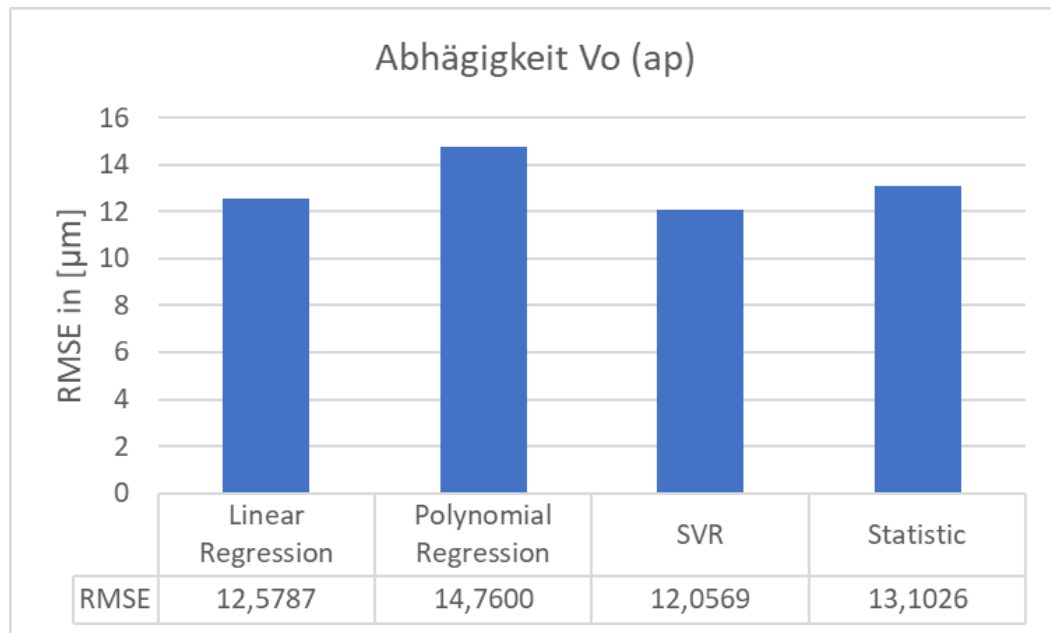


Abbildung 4-16: Validierungsergebnisse für die Abhängigkeit der Verschleißtiefe (Oben) von der axialen Schnitttiefe

Für die Funktion $V_o(ap)$ wurde der kleinste RMSE-Wert vom SVR-Algorithmus angezeigt (Abbildung 4-16). Der höchste RMSE-Wert wurde durch Polynomregression unter Verwendung des Konjugierte-Gradienten-Verfahrens gezeigt. Dies liegt daran, dass der Wert des Pearson-Koeffizienten größer als der Spearman-Korrelationskoeffizient ist. Das heißt, in dieser Funktion ist die lineare Korrelation stärker als die monotone.

Obwohl die Methode der kleinsten Quadrate den genauen Wert der optimalen Koeffizienten ermittelt, zeigte sie im Testdatensatz schlechtere Ergebnisse als die Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens und SVR. Dies beweist die Behauptung, dass eine bessere Annäherung für einen Trainingsdatensatz keine bessere Generalisierung der Daten bedeutet.

Für die Funktion $V_o(ap)$ wurde der SVR-Algorithmus für das adaptive Steuerungssystem und die Online-Überwachung des Werkzeugstatus ausgewählt.

Da der SVR-Algorithmus ein verstecktes Modell hat, wurde für theoretische Studien des Prozesses eine Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens als Alternative zur Berechnungsmethode der kleinsten Quadrate gewählt. Der Unterschied zwischen den Modellen selbst ist sehr gering, die mit diesen beiden Methoden erzielt wurden. Gleichzeitig ist das Konjugierte-Gradienten-Verfahrens flexibel und erfordert keine Änderungen, während das Modell kompliziert wird.

Der Fehler der ausgewählten Algorithmen, die auf den als Ergebnis der Experimente erhaltenen Datensatz trainiert wurden, ist für die gestellte Aufgaben zufriedenstellend. Um die Genauigkeit weiter zu verbessern, muss das Trainingsset ergänzt werden.

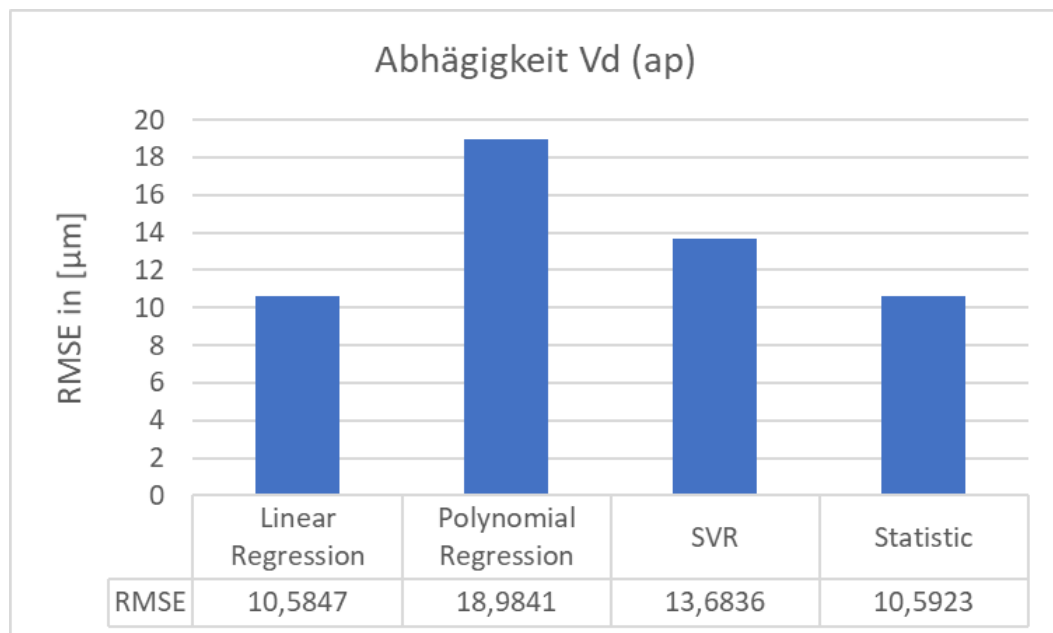


Abbildung 4-17: Validierungsergebnisse für die Abhängigkeit der Verschleißtiefe (Draufsicht) von der axialen Schnitttiefe

Für die Funktion $V_d(ap)$ wurde der kleinste RMSE-Wert von der Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens angezeigt (Abbildung 4-17). Der höchste RMSE-Wert wurde durch Polynomregression unter Verwendung des Konjugierte-Gradienten-Verfahrens gezeigt. Dies liegt daran, dass der Wert des Pearson-Koeffizienten größer als der Spearman-Korrelationskoeffizient ist. Das heißt, in dieser Funktion ist die lineare Korrelation stärker als die monotone.

Die Methode der kleinsten Quadrate und Lineare Regression haben fast gleiche Genauigkeit und erzielte Modelle in diesem Datensatz. Diesmal hat SVR-Algorithmus schlechtere Genauigkeit. Ein möglicher Grund ist fast maximale mögliche lineare Korrelation zwischen dem Parameter und der Zielfunktion.

Für diese Funktion wurde sowohl für praktische, als auch für wissenschaftliche Anwendungen die Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens ausgewählt.

Der erzielte Fehler des ausgewählten Algorithmus ist niedriger, als bei der Abhängigkeit $V_o(ap)$. Dieser Fehlerwert ist für die gestellte Aufgaben zufriedenstellend. Um die Genauigkeit weiter zu verbessern, muss das Trainingsset ergänzt werden.

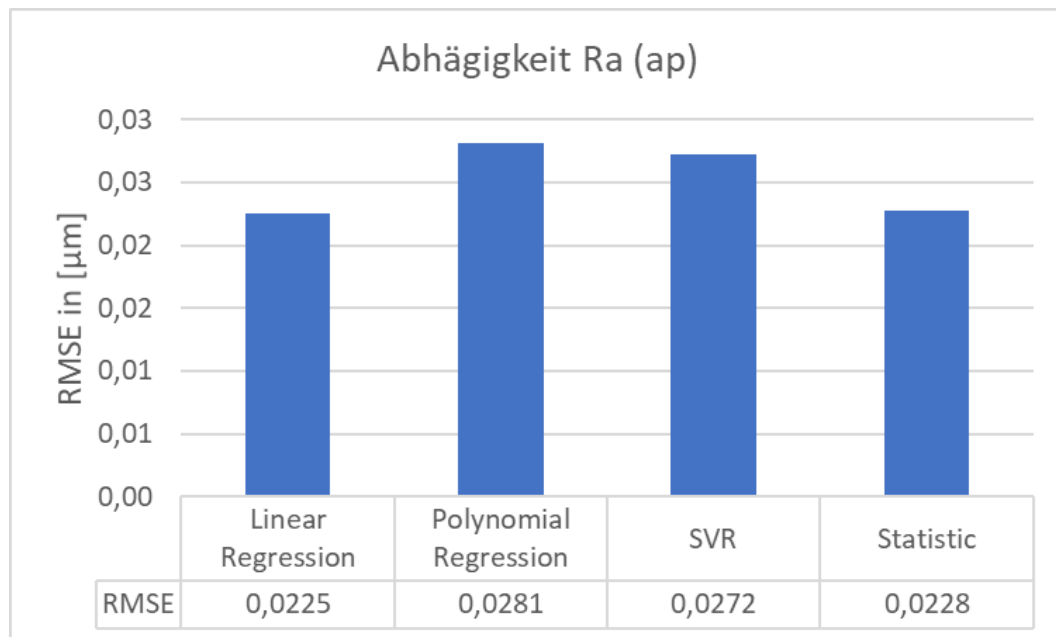


Abbildung 4-18: Validierungsergebnisse für die Abhängigkeit der Oberflächenrauheit von der axialen Schnitttiefe

Für die Funktion Ra (ap) wurde der kleinste RMSE-Wert von der Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens angezeigt (Abbildung 4-18). Der höchste RMSE-Wert wurde durch Polynomregression unter Verwendung des Konjugierte-Gradienten-Verfahrens gezeigt. Der Unterschied zwischen der Polynomregression und den linearen Lösungen ist in diesem Fall sehr gering. Dies liegt daran, dass der Wert des Spearman-Koeffizienten größer als der Pearson-Korrelationskoeffizient ist. Das heißt, in dieser Funktion ist die monotone Korrelation stärker als die lineare.

Ein möglicher Grund, warum lineare Lösungen haben trotzdem bessere Genauigkeit, ist die Möglichkeit, dass das Konjugierte-Gradienten-Verfahren für die Polynomregression lokale Minimum der Funktion gefunden hat.

Die Methode der kleinsten Quadrate und Lineare Regression haben fast gleiche Genauigkeit und erzielte Modelle in diesem Datensatz. Der SVR-Algorithmus hat schlechtere Genauigkeit.

Für diese Funktion wurde sowohl für praktische, als auch für wissenschaftliche Anwendungen die Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens ausgewählt.

Der erzielte Fehlerwert des ausgewählten Algorithmus ist für die gestellte Aufgaben zufriedenstellend. Um die Genauigkeit weiter zu verbessern, muss das Trainingsset ergänzt werden.

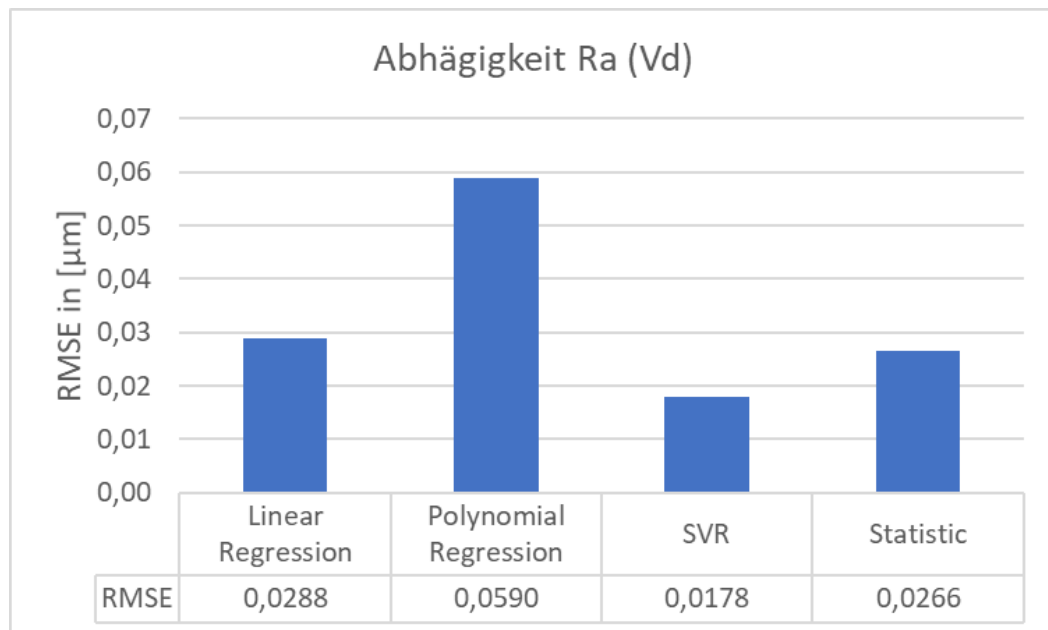


Abbildung 4-19: Validierungsergebnisse für die Abhängigkeit der Oberflächenrauheit von der Verschleißtiefe (Draufsicht)

Für die Funktion $Ra(V_d)$ wurde der kleinste RMSE-Wert vom SVR-Algorithmus angezeigt (Abbildung 4-19). Der wesentlich höhere RMSE-Wert wurde durch Polynomregression unter Verwendung des Konjugierte-Gradienten-Verfahrens gezeigt. Dies liegt daran, dass der Wert des Pearson-Koeffizienten größer als der Spearman-Korrelationskoeffizient ist. Das heißt, in dieser Funktion ist die lineare Korrelation stärker als die monotone. Die Methode der kleinsten Quadrate hat bessere Genauigkeit als die Lineare Regression.

Für die Funktion $Ra(V_d)$ wurde der SVR-Algorithmus für das adaptive Steuerungssystem und die Online-Überwachung des Werkzeugstatus ausgewählt.

Da der SVR-Algorithmus ein verstecktes Modell hat, wurde für theoretische Studien des Prozesses eine Lineare Regression mittels des Konjugierte-Gradienten-Verfahrens als Alternative zur Berechnungsmethode der kleinsten Quadrate gewählt. Obwohl die Methode der kleinsten Quadrate genauer in diesem Fall, ist das Konjugierte-Gradienten-Verfahrens flexibel und erfordert keine Änderungen, während das Modell kompliziert wird.

Der Fehler der ausgewählten Algorithmen, die auf den als Ergebnis der Experimente erhaltenen Datensatz trainiert wurden, ist für die gestellte Aufgaben zufriedenstellend. Um die Genauigkeit weiter zu verbessern, muss das Trainingsset ergänzt werden.

4.5 Klassifikationsalgorithmen

4.5.1 Logistische Regression mithilfe des Konjugierte-Gradienten-Verfahrens

Logistische Regression keine Regression ist, sondern ein Klassifizierungslernalgorithmus. Der Name stammt aus der Statistik und beruht auf der Tatsache, dass die mathematische Formulierung der logistischen Regression der Linearen Regression ähnlich ist [13]. Während die lineare Regression numerische Gleichungen behandelt und numerische Vorhersagen bildet, um Beziehungen zwischen Variablen zu erkennen, sagt die logistische Regression diskrete Klassen vorher [12].

Das Ziel der binären Klassifizierung besteht darin, eine Funktion zu erhalten, die die beiden Klassen trennen kann. Klassen werden in binärer Form geschrieben, d.h. eine Klasse wird als negative Bezeichnung 0 und die zweite als positive Bezeichnung 1 markiert.

Das bedeutet, dass eine einfache stetige Funktion gefunden werden muss, deren Ausgabewerte im Intervall (0,1) liegen. In diesem Fall wird x eine negative Bezeichnung zugewiesen, wenn der vom Modell für Eingabe x zurückgegebene Wert näher an 0 liegt. Andernfalls wird das Beispiel als positiv markiert. Eine der Funktionen mit dieser Eigenschaft ist die Standardlogistikfunktion, auch als Sigmoidfunktion bzw. S-Funktion (Formel 4-19) bekannt [13].

$$f(z) = \frac{1}{1+e^{-z}} \quad (4-19)$$

Die Grafik dieser Funktion ist in der Abbildung 4-20 dargestellt:

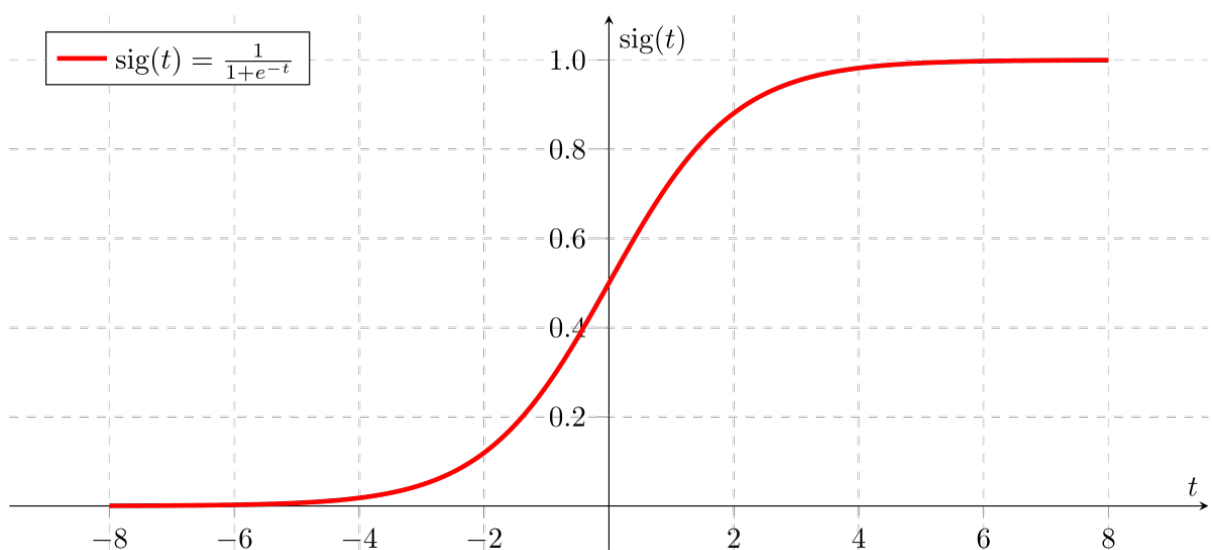


Abbildung 4-20: Grafik der S-Funktion [62]

Aus dem Diagramm der Standardlogistikfunktion kann man ansehen, wie gut es zu dem gestellten Klassifizierungsziel passt: Die Ausgabe von $f(x)$ kann man als die Wahrscheinlichkeit interpretieren, dass y positiv ist. Wenn es beispielsweise höher oder gleich dem Schwellenwert 0,5 ist, kann man sagen, dass die Klasse von x positiv ist. Andernfalls ist es negativ. In der Praxis kann die Wahl des Schwellenwerts je nach Problem unterschiedlich sein. Wenn z gegen unendlich geht, wird $f(z)$ zu 1 und wenn z gegen negativen unendlich geht, wird $f(z)$ zu 0.

In der Logistischen Regression ist das Ziel die Zielgröße y immer noch als lineare Funktion von x zu modellieren. Die Zielgröße y soll dabei binär sein. Die Standardlogistikfunktion ermöglicht die lineare Funktion wie $w \times x + b$, die von minus unendlich bis plus unendlich reicht, konvertieren und in einen numerischen Wert zwischen 0 und 1 abzubilden, ohne jedoch diese genauen Grenzen zu erreichen. Das logistische Regressionsmodell sieht also folgenderweise aus:

$$f_{w,b}(x) = \frac{1}{1 + e^{-(w \times x + b)}} \quad (4-20)$$

Die Ausgabe des Modells ist in diesem Fall nicht direkt der vorhergesagte Wert, sondern die geschätzte Wahrscheinlichkeit. Dies wird verwendet, um zu schließen, wie sicher der vorhergesagte Wert der tatsächliche Wert sein kann, wenn eine Eingabe x gegeben wird. Mathematisch kann dies geschrieben werden als:

$$f_{w,b}(x) = P(y=1 | x; w, b) \quad (4-21)$$

$f_{w,b}(x)$ ist die Wahrscheinlichkeit, dass $y=1$ ist, gegebenes x , der durch w und b parametrisiert ist. Da es nur zwei Klassen gibt, folgt daraus:

$$\begin{aligned} P(y=1 | x; w, b) + P(y=0 | x; w, b) &= 1 \\ P(y=0 | x; w, b) &= 1 - P(y=1 | x; w, b) \end{aligned} \quad (4-22)$$

Aus der Grafik der S-Funktion geht hervor, dass Schwellenwert standardmäßig 0,5 beträgt. Die Klassifizierungsfunktion wird so aussehen:

$$f_{w,b}(x) = \begin{cases} \rightarrow 1, & \text{wenn } \frac{1}{1 + e^{-(w \times x + b)}} \geq 0,5 \\ \rightarrow 0, & \text{wenn } \frac{1}{1 + e^{-(w \times x + b)}} < 0,5 \end{cases} \quad (4-23)$$

Aus der Grafik der S-Funktion geht es auch hervor, dass $f_{w,b}(x) \geq 0,5$, wenn $w \times x + b \geq 0$. Deshalb kann man die Klassifizierungsfunktion aus der Formel 4-24 folgenderweise überschreiben:

$$f_{w,b}(x) = \begin{cases} \rightarrow 1, & \text{wenn } w \times x + b \geq 0 \\ \rightarrow 0, & \text{wenn } w \times x + b < 0 \end{cases} \quad (4-24)$$

Die Lernaufgabe besteht hier wie bei der Linearen Regression darin, die optimale Werte für Parameter w und b zu kriegen.

Die lineare Regression verwendet den mittleren quadratischen Fehler als Kostenfunktion. Wenn solche Kostenfunktion für die logistische Regression verwendet wird, handelt es sich um eine nicht konvexe Funktion von Parametern. Der Gradientenabstieg konvergiert nur dann zum globalen Minimum, wenn die Funktion konvex ist [13].

Deshalb sieht die Kostenfunktion für die Logistische Regression anders aus. Die Gleichung ist in der Formel 4-25 angegeben. Die Kostenfunktion ist als J bezeichnet.

$$J_{w,b} = -\frac{1}{N} \times \left[\sum_{i=1}^N y_i \times \log(f_{w,b}(x_i)) + (1-y_i) \times \log(1-f_{w,b}(x_i)) \right] \quad (4-25)$$

Der Lernalgorithmus soll iterativ solche Werte von w und b finden, bei denen die Kostenfunktionswert minimal ist. Dafür wird wie bei der Linearen Regression das Konjugierte-Gradienten-Verfahren als Optimierungsverfahren eingesetzt.

Als erstes soll der Gradient bzw. der Richtungsvektor bestimmt werden, der aus partiellen Ableitungen besteht (Formel 4-26).

$$\nabla J_{w,b} \equiv \left[\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b} \right] \quad (4-26)$$

Die Parameteränderung wird dementsprechend in jeder Iteration nach Formel 4-27 berechnet:

$$\begin{aligned} \Delta w &= \eta \times \frac{\partial J}{\partial w} \\ \Delta b &= \eta \times \frac{\partial J}{\partial b} \end{aligned} \quad (4-27)$$

Die partiellen Ableitungen sind dabei:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{w}} &= \frac{1}{N} \times \sum_{i=1 \dots N} (f_{\mathbf{w},\mathbf{b}}(\mathbf{x}_i) - y_i) \times \mathbf{x}_i \\ \frac{\partial J}{\partial \mathbf{b}} &= \frac{1}{N} \times \sum_{i=1 \dots N} (f_{\mathbf{w},\mathbf{b}}(\mathbf{x}_i) - y_i)\end{aligned}\quad (4-28)$$

Man kann bemerken, dass die partiellen Ableitungen bei der Logistischen Regression genau wie bei der Linearen Regression aussehen. Der Unterschied besteht hier aber in der Funktion $f_{\mathbf{w},\mathbf{b}}(\mathbf{x})$, die nicht linear, sondern S-Funktion ist.

Die Aktualisierung der Parameter soll iterativ wiederholen, bislang die Funktion ein Minimum erreicht (Formel 4-6). Alle Parameter während der Iteration müssen gleichzeitig aktualisiert werden.

Dieser Algorithmus wurde mit einer Programmiersprache implementiert. Die Eingabe in den Algorithmus ist ein Datensatz und Hyperparameterwerte. Die Hyperparameter dieses Algorithmus sind konventionell Lernrate und Anzahl der Iterationen. Die Auswahl der Anzahl der Iterationen, um das Minimum zu erreichen, nimmt Zeit in Anspruch und erschwert das Einrichten des Algorithmus. Deshalb wurde in dem entwickelten Programm zur Steigerung der Produktivität der Algorithmus so modifiziert, dass die Anzahl der Iterationen automatisch bestimmt wird. Der Algorithmus stoppt automatisch, wenn das Minimum der Kostenfunktion erreicht ist. Unten ist eine Stoppbedingung für den Algorithmus angegeben:

$$J_{\mathbf{w},\mathbf{b}}(j) - J_{\mathbf{w},\mathbf{b}}(j-1) \leq t \quad (4-29)$$

In diesem Formel Ausdruck bezeichnet j die Iterationsnummer des Algorithmus und t ist die Toleranz ($t = 0,000001 \dots 0,0001$).

Darüber hinaus wurde die Kostenfunktion in zwei Teile geteilt, weil in manchen Fällen Null unter dem Logarithmus auftritt. Solcherweise kann man diese Situation vermeiden.

Da der Lernprozess im Vergleich zu der Linearen Regression nicht verändert wird, kann man ihn auch zu Diagnosezwecken anhand von zwei Diagrammen (Abbildung 4-21 und Abbildung 4-22) visualisieren.

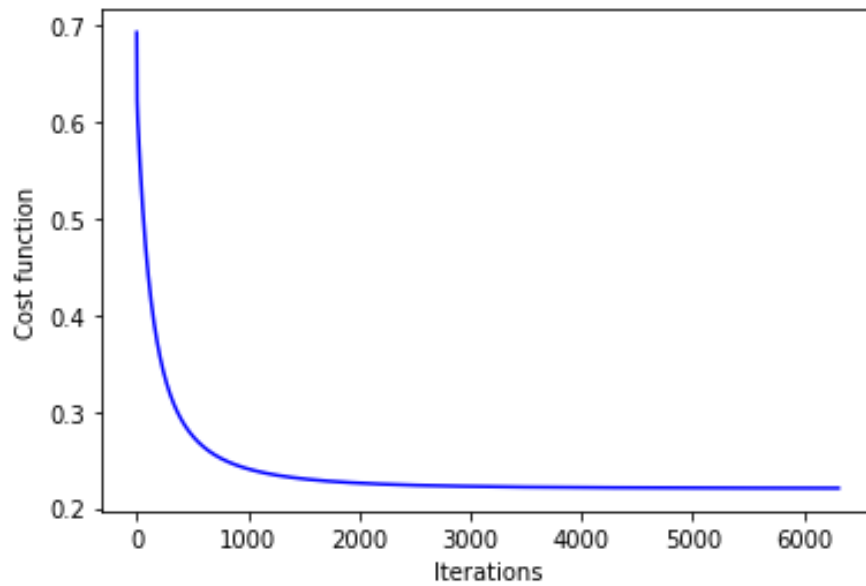


Abbildung 4-21: Die Veränderung von dem Kostenfunktionswert im Laufe des Lernens

Abbildung 4-21 zeigt, wie sich der Wert von der Kostenfunktion während des Trainings ändert. Mit dieser Grafik kann man feststellen, ob der Lernprozess gut verläuft. Wenn der Kostenfunktionswert mit jeder Iteration abnimmt, ist der Lernprozess erfolgreich. Andernfalls findet der Lernprozess nicht statt. Dies kann auf einen Fehler im Code oder einen großen Lernrate-Wert zurückzuführen sein.

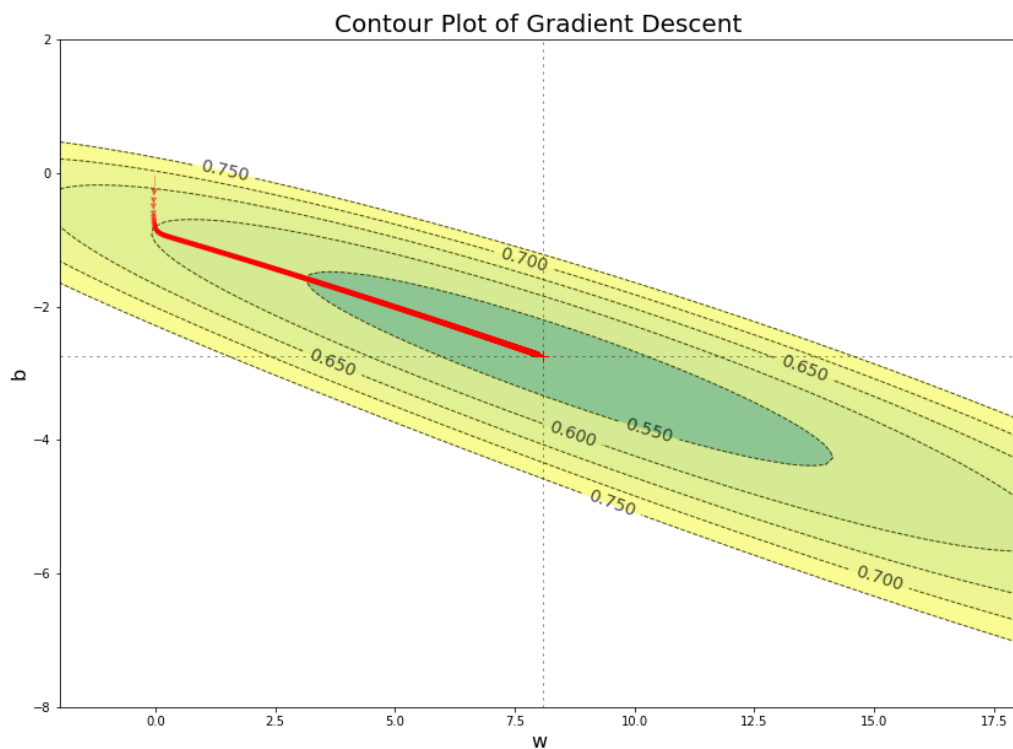


Abbildung 4-22: Konturdiagramm von Gradientenabstieg

Das Konturdiagramm des Gradientenabstiegs zeigt die Änderung der Werte der Gewichtskoeffizienten und der Position des Wertes der Funktion relativ zu ihrem Minimum nach dem Training. Auf diese Weise kann festgestellt werden, ob die Funktion ein Minimum erreicht hat.

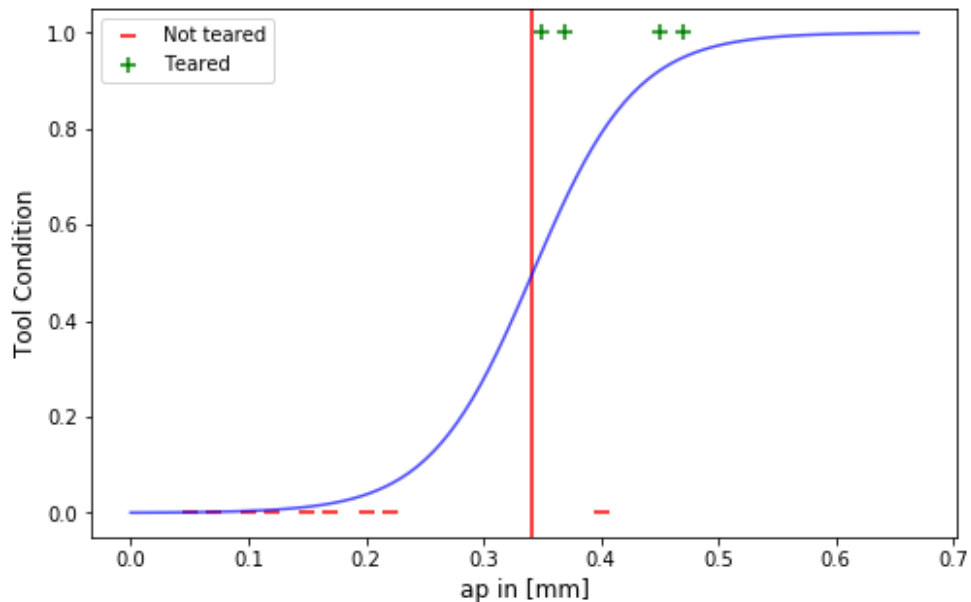


Abbildung 4-23: Beispiel des Trainingsergebnisses bei der Logistischen Regression

Abbildung 4-23 zeigt das Ergebnis des Lernens des Algorithmus. Die als Ergebnis des Experiments erhaltenen Daten sind in der Grafik mit „+“ (verschließen) und mit „-“ (nicht verschließen) markiert. Die als Ergebnis des Trainings erhaltene S-Funktion wird durch eine blaue Linie angezeigt. Die rote Linie ist die Entscheidungsgrenze zwischen zwei Klassen. Alle Beispiele links werden als „-“ von dem Algorithmus klassifiziert. Alle Beispiele rechts werden als „+“ klassifiziert.

Nach dem Training wurde dieses Algorithmus durch das „4-fold cross-validation“-Verfahren validiert. Als Validierungskriterien wurden „Accuracy“ und „F1-Score“ ausgewählt. Für die Vorhersage des Werkzeugzustands wird nur die axiale Schnitttiefe betrachtet. Die Ergebnisse sind in Tabelle 4-8 angegeben.

Tabelle 4-8: Ergebnisse des „4-fold cross-validation“-Verfahrens

Kriterium	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Gesamtwert
Accuracy	0,88	0,94	0,8	0,92	0,89
F1-Score	0,88	0,94	0,72	0,9	0,86

Nach dem „4-fold cross-validation“ wurden alle Daten fürs Lernen benutzt und das folgende Modell erzielt (Tabelle 4-9):

Tabelle 4-9: Erzieltes Modell nach dem Lernen mittels Logistische Regression

Algorithm	Modell
Logistische Regression	Tool condition= $\frac{1}{1+e^{-(19,22 \times a_p - 5,02)}}$

4.5.2 Neurale Netzwerke

Künstliche neuronale Netze, auch als neuronale Netze bekannt, sind eine übliche ML-Technik zur Verarbeitung von Daten durch Analyseebenen. Die Benennung künstlicher neuronaler Netze wurde durch die Ähnlichkeit des Algorithmus mit dem menschlichen Gehirn inspiriert [12].

Ähnlich wie Neuronen im menschlichen Gehirn werden künstliche neuronale Netze durch miteinander verbundene Neuronen gebildet, die auch als Knoten bezeichnet werden und über Axone, sogenannte Kanten, miteinander interagieren. In einem neuronalen Netzwerk sind die Knoten in Schichten gestapelt und beginnen im Allgemeinen mit einer breiten Basis. Die erste Schicht besteht aus Rohdaten wie numerischen Werten, Text, Bildern oder Ton, die in Knoten unterteilt sind. Jeder Knoten sendet dann Informationen über die Netzwerkkanten an die nächste Knotenschicht [12].

Jede Kante hat ein numerisches Gewicht, das je nach Erfahrung geändert und formuliert werden kann. Wenn die Summe der verbundenen Kanten einen festgelegten Schwellenwert erfüllt, der als Aktivierungsfunktion bezeichnet wird, wird auf der nächsten Schicht ein Neuron aktiviert. Wenn jedoch die Summe der verbundenen Kanten den eingestellten Schwellenwert nicht erreicht, wird die Aktivierung nicht ausgelöst. Dies führt zu einer Alles-oder-Nichts-Anordnung.

Die Gewichte entlang jeder Kante einzigartig sind, um sicherzustellen, dass die Knoten unterschiedlich ausgelöst werden und nicht alle das gleiche Ergebnis liefern.

Um das Netzwerk durch überwachtes Lernen zu trainieren, wird der vorhergesagte Wert des Modells mit dem tatsächlichen Wert (die nach den Experimenten erzeugt werden) verglichen, und die Differenz zwischen diesen beiden Ergebnissen wird gemessen und als Kosten oder Kostenwert bezeichnet. Kostenwert wird als J bezeichnet.

$$J = -\frac{1}{N} \times \sum_{i=1}^N o(x_i) - y_i \quad (4-30)$$

Der Zweck des Trainings besteht darin, den Kostenwert zu reduzieren, bis die Vorhersage des Modells genau mit der korrekten Ausgabe übereinstimmt. Dies wird erreicht, indem die Gewichte des Netzwerks schrittweise angepasst werden, bis der niedrigstmögliche Kostenwert erreicht wird.

Dieser Prozess des Trainings des neuronalen Netzwerks wird als Backpropagation bezeichnet. Anstatt von links nach rechts zu navigieren, wie Daten in ein neuronales Netzwerk eingespeist werden, erfolgt die Rückausbreitung in umgekehrter Reihenfolge und verläuft von der Ausgabeschicht von rechts zur Eingabeschicht links [12].

Die allgemeine Struktur des neuronalen Netzwerks ist in Abbildung 4-24 dargestellt.

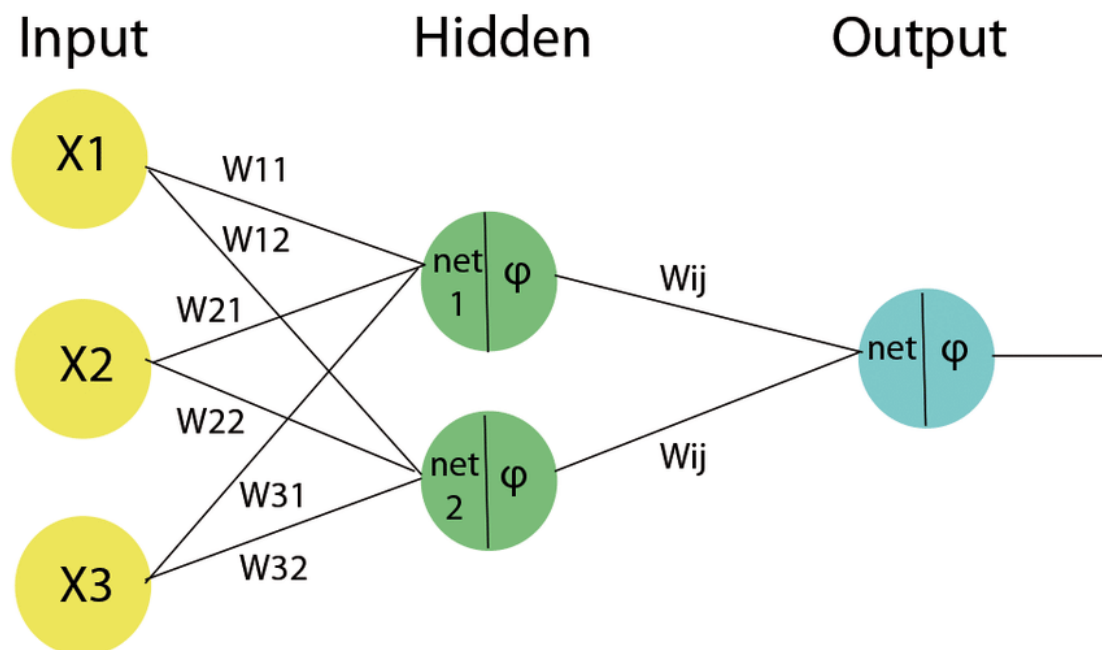


Abbildung 4-24: Struktur des neuronalen Netzwerks [63]

Die Arbeit neuronaler Netze wird durch die Anzahl der verborgenen Schichten, die Art der Neuronen sowie die Art der Verbindung zwischen Neuronen bestimmt.

Die Komplexität der Quelldaten bestimmt die Anzahl der notwendigen verborgenen Schichten. Die gestellte Aufgabe, die der Algorithmus ausführen wird, bestimmt die Art der Neuronen. Verschiedene Arten von Neuronen unterscheiden sich voneinander nur durch ihre Aktivierungsfunktion.

Für binäre Klassifizierungsprobleme werden üblicherweise Perzeptrone und Sigmoidneuronen verwendet, die eine Schritt- oder eine S-Funktion enthalten. In dieser Arbeit wurden Sigmoidneuronen ausgewählt, um ein neuronales Netzwerk aufzubauen.

Die Klassen in dem Datensatz, die nach Durchführung der Experimente erhalten wurden, sind linear trennbar. In diesem Fall zeigt der Aufbau eines neuronalen Netzwerks beliebiger Komplexität die gleichen Ergebnisse wie ein neuronales Netzwerk ohne verborgene Schichten [13].

Da sich die Komplexität des neuronalen Netzwerks negativ auf die Produktivität seines Trainings auswirkt, wurde in dieser Arbeit ein Design ohne verborgene Schichten gewählt. Das Design des entwickelten neuronalen Netzwerks ist in Abbildung 4-25 geschildert:

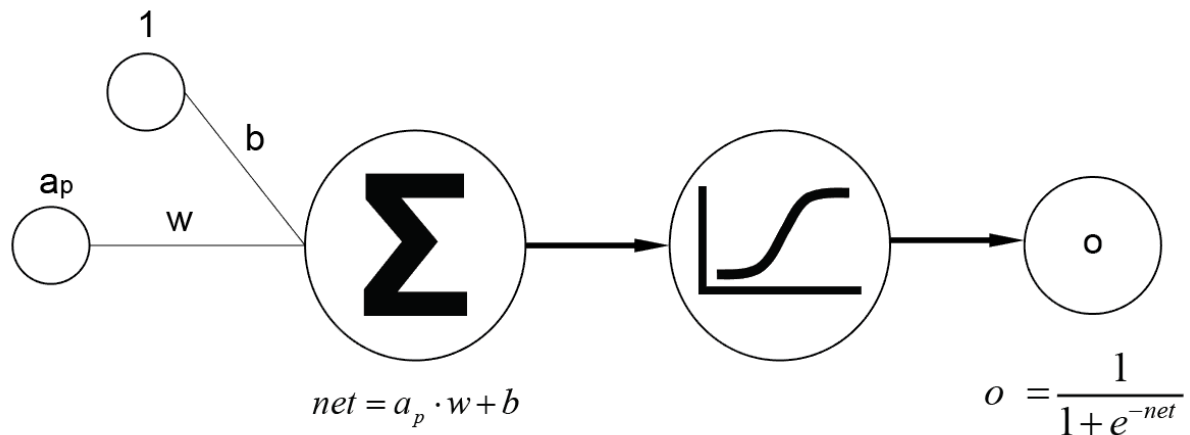


Abbildung 4-25: Neuronales Netz zur Überwachung des Werkzeugzustands

Einer der Nachteile neuronaler Netze besteht darin, dass sie als „Black-Box“ funktionieren. Das Netzwerk kann zwar genaue Ergebnisse annähern, die Verfolgung seiner Struktur jedoch nur begrenzte oder keine Einblicke in die Variablen liefert, die sich auf das Ergebnis auswirken.

Da dieses neuronale Netzwerk keine verborgenen Schichten aufweist und aus einem Sigmoid-Neuron besteht, sehen der Prozess und das Ergebnis seines Trainings genauso aus wie bei der logistischen Regression (Abbildungen 4-22 und 4-23).

Nach dem Training wurde dieses Algorithmus durch das „4-fold cross-validation“-Verfahren validiert. Als Validierungskriterien wurden „Accuracy“ und „F1-Score“ ausgewählt. Für die Vorhersage des Werkzeugzustands wird nur die axiale Schnitttiefe betrachtet. Die Ergebnisse sind in Tabelle 4-10 angegeben.

Tabelle 4-10: Ergebnisse des „4-fold cross-validation“-Verfahrens

Kriterium	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Gesamtwert
Accuracy	0,88	0,94	0,8	0,92	0,89
F1-Score	0,88	0,94	0,72	0,9	0,86

Neurales Netzwerk hat gleiche Ergebnisse, wie die Logistische Regression gezeigt, weil in diesem Fall es genauso wie sie gelernt habe. Mit Steigerung der Komplexität werden die Leistungen unterschiedlich.

4.5.3 Support Vector Machines

Support Vector Machines (SVM) ist ein ML-Algorithmus, der für verschiedene Aufgaben verwendet werden kann. In dieser Arbeit wird er als Klassifikationsalgorithmus betrachtet. Das Hauptziel bei SVM besteht darin, die optimale Hyperebene zu finden, um zwischen Datenpunkten verschiedener Klassen korrekt zu klassifizieren [13].

Hyperebenen sind Entscheidungsgrenzen, mit denen die Datenpunkte klassifiziert werden können. Datenpunkte, die auf beide Seiten der Hyperebene fallen, können verschiedenen Klassen zugeordnet werden. Um die beiden Klassen von Datenpunkten zu trennen, können viele mögliche Varianten der Hyperebenen ausgewählt werden. Das ist in der Abbildung 4-26 (a) geschildert.

Die Logistische Regression teilt die Datenpunkte so in zwei Klassen auf, dass der Abstand zwischen allen Datenpunkten und der Hyperebene minimiert wird. Somit wird der Algorithmus für jeden Punkt im Datensatz, einschließlich Außenreiser, sehr empfindlich. Das wirkt sich anschließend negativ auf die Genauigkeit der Klassifizierung neuer Daten aus. Dies liegt daran, dass Punkte in der Nähe einer realen Hyperebene falsch klassifiziert werden [13].

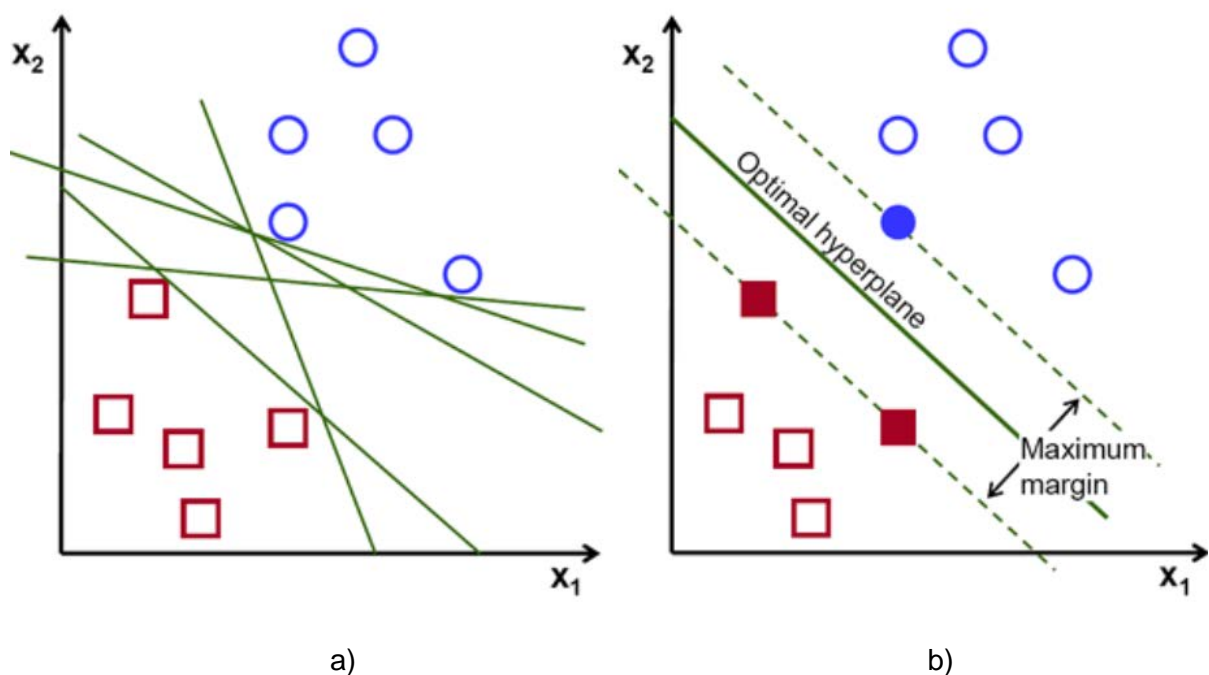


Abbildung 4-26: Mögliche varianten der Hyperebenen [64]

SVM versucht, den Abstand zwischen den Unterstützungsvektoren zu maximieren (Abbildung 4-26 (b)). Die Unterstützungsvektoren sind die nächstgelegenen Punkte von Vertretern verschiedener Klassen, die der Hyperebene am nächsten liegen. Sie sind in Abbildung 4-27 dargestellt.

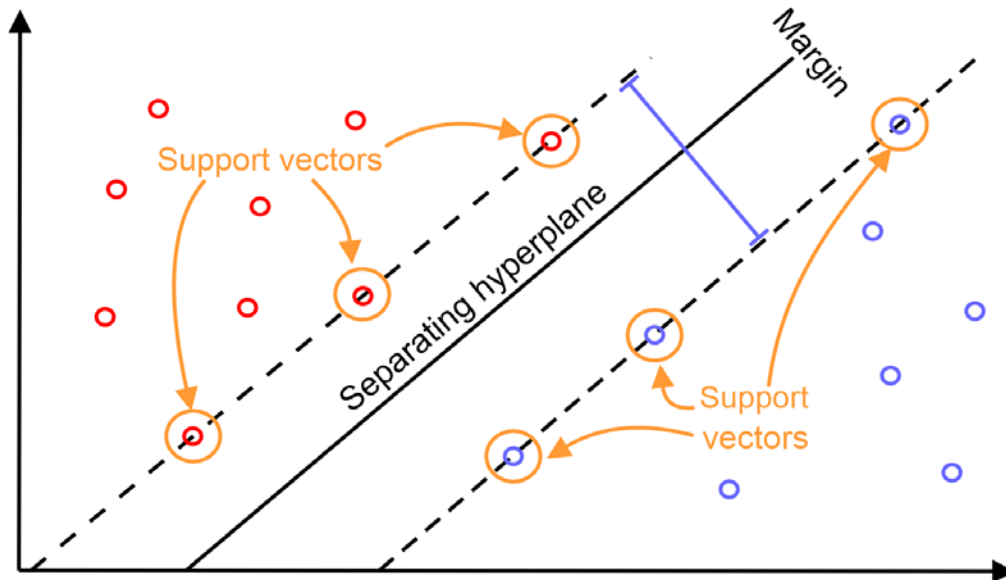


Abbildung 4-27: Unterstützungsvektoren bei den Support Vector Machines [65]

SVM basiert auf geometrischen Eigenschaften der Daten, während die logistische Regression auf statistischen Ansätzen basiert. Die logistische Regression erzeugt probabilistische Werte zwischen 0 und 1, während SVM nur diskrete Werte 1 oder 0 erzeugt. Deshalb sieht die SVM-Funktion, wie eine Schrittfunktion:

$$f_{w,b}(x) = \begin{cases} 1, & \text{wenn } w \cdot x + b \geq 0 \\ 0, & \text{wenn } w \cdot x + b < 0 \end{cases} \quad (4-31)$$

Die Kostenfunktion bei SVM, deren Wert der SVM Algorithmus während des Trainings minimiert, ist in der Formula 4-32 angegeben.

$$J_{w,b} = C \times \left[\sum_{i=1}^N y_i \times \text{cost}_1(f_{w,b}(x_i)) + (1 - y_i) \times \text{cost}_0(f_{w,b}(x_i)) \right] \quad (4-32)$$

Man kann bemerken, dass die Kostenfunktionen für Logistische Regression und für SVM ähnlich sind. Der Hauptunterschied besteht in der Verlustfunktionen. SVM minimiert den Scharnierverlust, während die logistische Regression den logistischen Verlust minimiert. Der Scharnierverlust wird nach Formel 4-33 berechnet:

$$\text{cost}(f(x)) = \begin{cases} \max(0, 1 - (w \times x + b)), & \text{wenn } y = 1 \\ \max(0, 1 + (w \times x + b)), & \text{wenn } y = 0 \end{cases} \quad (4-33)$$

Die Kostenfunktion des Algorithmus enthält ein Hyperparameter C. Er dient zur Regulierung der Einschränkungen. C-Parameter priorisiert, wie wichtig es ist, den Hyperebenen für Trainingsdaten zu optimieren und bewirkt Abstand zu Unterstützungsvektoren [13].

Mit einem sehr großen Wert von C ist dieser Klassifikator ähnlich wie bei Logistischen Regression empfindlich gegenüber Ausreißern. Der Abstand zu Unterstützungsvektoren ist dabei klein.

Dieser Algorithmus wurde in der Entwicklungsumgebung realisiert. Die Eingabe in den Algorithmus ist ein Datensatz und Hyperparameterwert C.

Support Vector Machine Algorithmus gehört auch zu den sogenannten „Black Box“-Algorithmen [54]. In diesem Fall ist es unmöglich sowohl den Lernprozess zu überwachen, als auch die Ergebnisse zu Interpretieren. Dieser Algorithmus gibt kein Model und keine Wahrscheinlichkeit, dass ein Beispiel zu den Klassen gehört. Deshalb ist es möglich die Leistung des gelernten Algorithmus nur anhand der Ergebnisse zu evaluieren.

Das Ergebnis des Lernens des Algorithmus bei Support Vector Machine ist in der Abbildung 4-28 angezeigt. Da der Algorithmus die Klassifikation ohne S-Funktion macht, wird als Ergebnis nur die Entscheidungsgrenze geschildert. Die Entscheidungsgrenze wird als rote vertikale Linie geschildert.

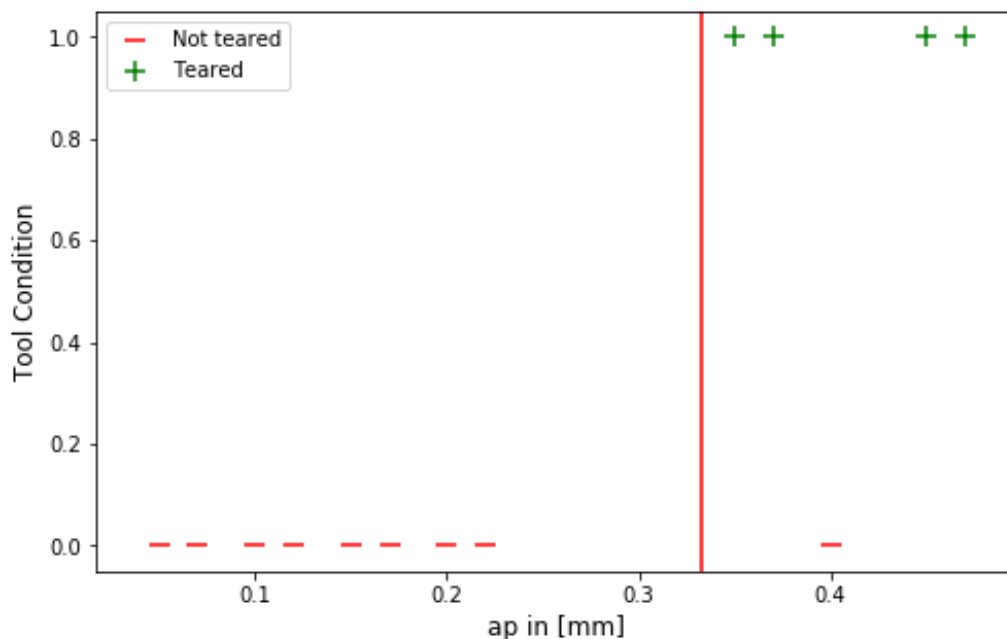


Abbildung 4-28: Beispiel des Trainingsergebnisses bei Support Vector Machine

Nach dem Training wurde dieses Algorithmus durch das „4-fold cross-validation“-Verfahren validiert. Als Validierungskriterien wurden „Accuracy“ und „F1-Score“ ausgewählt. Fürs Lernen wurde Hyperparameter C=10 gewählt. Für die Vorhersage des Werkzeugzustands wird nur die axiale Schnitttiefe betrachtet. Die Ergebnisse sind in Tabelle 4-11 angegeben.

Tabelle 4-11: Ergebnisse des „4-fold cross-validation“-Verfahrens

Kriterium	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Gesamtwert
Accuracy	0,97	0,89	0,84	0,92	0,91
F1-Score	0,97	0,90	0,85	0,9	0,91

4.6 Auswahl der Klassifikationsalgorithmen

Nach der Entwicklung und Validierung von Algorithmen zur Klassifizierung des Werkzeugzustand während des Drehfräsens von CoCrMo-Legierungen muss derjenige ausgewählt werden, der die größte Genauigkeit aufweist.

Diese Arbeit sieht die weitere Verwendung von Algorithmen für theoretische und praktische Zwecke vor. Deshalb wurde auch berücksichtigt, in welcher Form und wie viele Informationen der Algorithmus am Ausgang liefert.

Zur Beurteilung der Genauigkeit der Klassifizierungsalgorithmen wurden zwei Kriterien verwendet: „Accuracy“ und „F1-Score“. Das Kriterium „Accuracy“ ist sehr einfach zu interpretieren und ein klares Kriterium für die Genauigkeit des Algorithmus. Es zeigt, wie viele Beispiele aus der Gesamtzahl korrekt klassifiziert wurden.

„Accuracy“ kann verwendet werden, wenn die Klassenverteilung gleichmäßig ist, während der „F1-Score“ eine bessere Metrik ist, wenn unausgeglichene Klassen vorhanden sind. Bei den meisten realen Klassifizierungsproblemen besteht eine unausgewogene Klassenverteilung, und daher ist es empfehlenswert noch den „F1-Score“ als Metrik für die Bewertung des Modells zu benutzen.

Alle Ergebnisse wurden in Diagrammen dargestellt. Zur eindeutigen Interpretation der Ergebnisse sind unten die genauen Werte für „Accuracy“ und „F1-Score“ für alle Algorithmen angegeben.

Abbildung 4-29 zeigt den Gesamtwert für „Accuracy“ aller entwickelten Algorithmen. Die beste durchschnittliche Genauigkeit bei Testdaten wird durch den Support Vector Machines-Algorithmus demonstriert. Die Genauigkeit der Logistischen Regression und des Neuralen Netzwerks ist jedoch nicht wesentlich schlechter und unterscheidet sich nur um 2%.

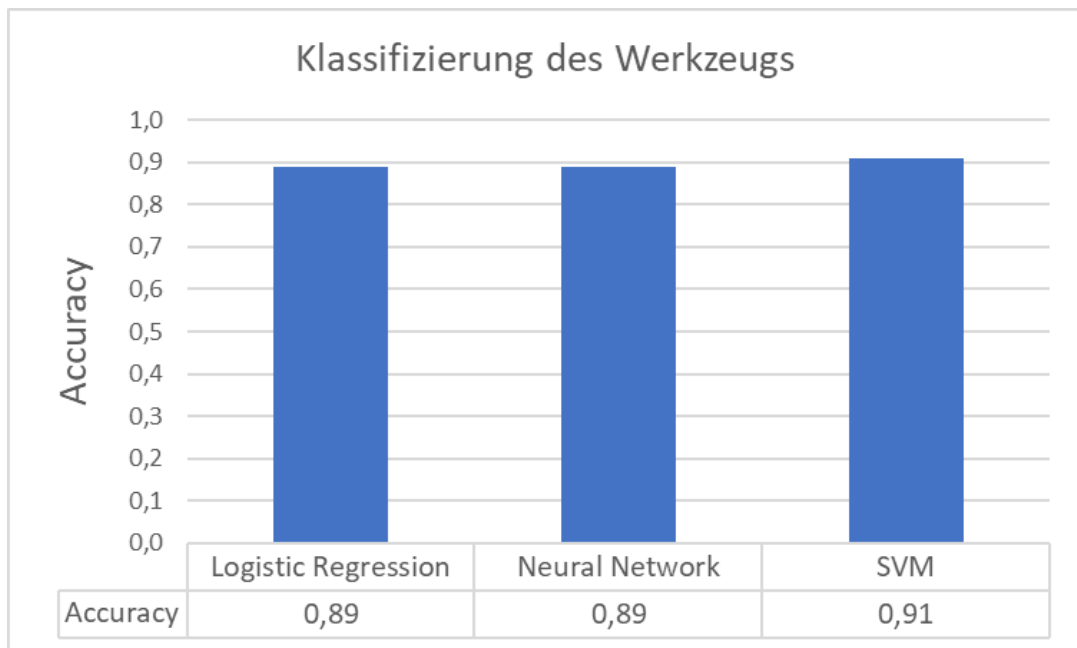


Abbildung 4-29: „Accuracy“ der Klassifizierungsalgorithmen

Abbildung 4-30 zeigt den gesamten „F1-Score“ aller entwickelten Algorithmen. Man kann feststellen, dass sich die Werte für „F1-Score“ und „Accuracy“ nicht wesentlich unterscheiden. Es kann gefolgert werden, dass die gesammelten Daten gleichmäßig über die Klassen verteilt sind. Die Ergebnisse von "F1-Score" bestätigen nur die Ergebnisse, die zuvor unter Verwendung des Kriteriums „Accuracy“ erhalten wurden.

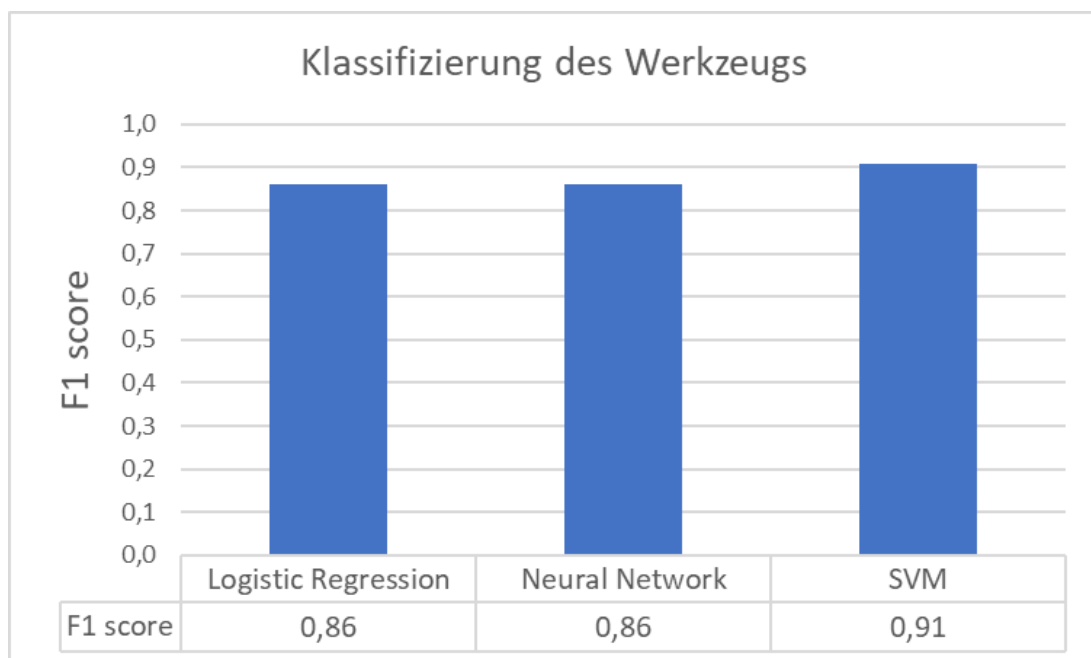


Abbildung 4-30: „F1-Score“ der Klassifizierungsalgorithmen

Für den praktischen Einsatz in Produktionsprozessen wurde der Support Vector Regression-Algorithmus gewählt, weil er die höchste Genauigkeit zeigte. Er kann jedoch nicht für

theoretische Anwendungen verwendet werden, da er keine Informationen über das Modell liefert. Für theoretische Zwecke wurde die Logistische Regression als Algorithmus ausgewählt, da er gleiche Genauigkeit wie das Neuronale Netzwerk hat und ein Modell zur Klassifizierung erzeugt.

4.7 Anwendungsschemen der Algorithmen und Modelle

4.7.1 Schema zur Optimierung der wissenschaftlichen Forschungen

Der wissenschaftliche Prozess kann im Allgemeinen durch einen Kreislauf dargestellt werden (Abbildung 4-31). Anhand der theoretischen Kenntnisse oder Modelle, stellen die Wissenschaftler neue Hypothesen auf. Dann folgen die Experimente für die Überprüfung der Gültigkeit von Hypothesen. Die Versuchsergebnisse können die Hypothesen bestätigen oder nicht bestätigen, die Grundlage für neue Hypothesen bilden und auf ihrer Basis können auch Modelle erstellt werden.

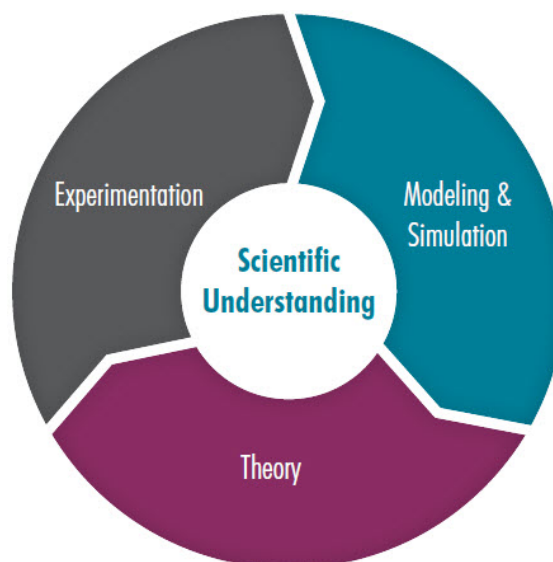


Abbildung 4-31: Rolle der Modellierung und Simulation bei der wissenschaftlichen Entdeckung [66]

Modelle ermöglichen die Simulationsdurchführung mithilfe der Programmierung oder vorhandenen Softwarelösungen. Dann beginnt der Zyklus erneut. Die erworbenen Kenntnisse und Modelle können für die weitere Forschungen verwendet werden.

In der Fertigungstechnik sind die Experimente in der Regel mit dem Zeit- und Kostenaufwand verbunden. Dazu gehört Einkauf der notwendigen Ausrüstung (Sensoren, Maschinen, Messgeräte), Rohteile, Werkzeuge usw. Deshalb wäre es sinnvoll, den wissenschaftlichen Prozess zu optimieren, um die benötigte Versuchsanzahl zu verringern und schneller Kenntnisse und Modelle zu kriegen.

Je weniger theoretische Grundlagen vorhanden sind, desto mehr Hypothesen im Rahmen der Forschung überprüft werden sollen. z.B. Welche Parameter den Prozess beeinflussen und wie. Das erhöht die Experimentanzahl für Modellerstellung.

Das Problem dabei besteht darin, dass moderne Bearbeitungsprozesse nicht so gut untersucht sind, und dementsprechend es gibt geringe Menge der aposteriorischen Kenntnisse. In dieser Arbeit wurde orthogonales Drehfräsen betrachtet, das ein Beispiel für solches Verfahren ist.

Konventionell bei der Modellerstellung werden die Beziehungen zwischen den gesammelten Daten mithilfe der Statistik bestimmt. Dann anhand dieser Ergebnisse und der Kenntnisse in der Fertigungstechnik werden die Modelle manuell erstellt und durch Programmierbefehle beschrieben (Abbildung 4-32). Das nimmt viel Zeit in Anspruch, diese Modelle können fehlerhaft sein und solche Programme sind nicht flexibel.

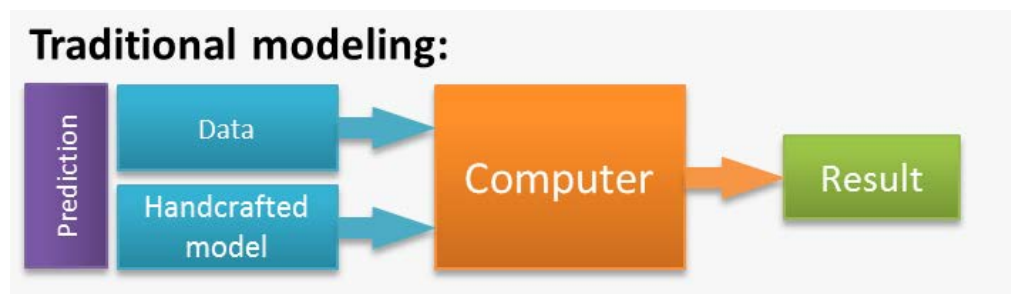


Abbildung 4-32: Traditioneller Modellierungsvorgang in der Wissenschaft [67]

Deshalb wurde auf Basis der entwickelten Algorithmen neues Schema für Modellierung in der Wissenschaftlichen Forschungen konzipiert (Abbildung 4-33).

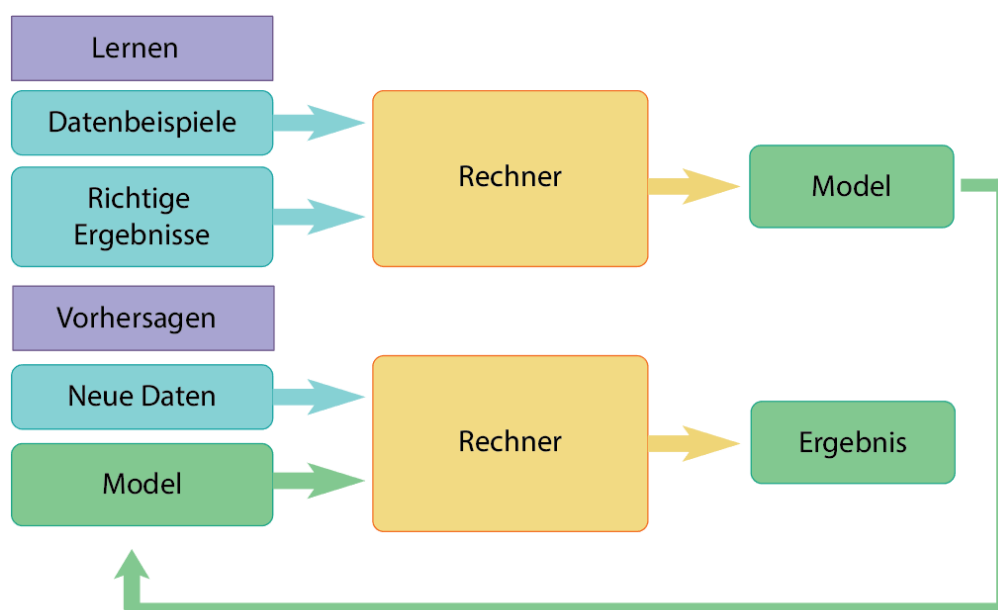


Abbildung 4-33: Konzipiertes Schema für die Modellerstellung

In diesem Schema werden anstatt der Statistik die entwickelten Machine Learning Algorithmen eingesetzt. Diese Algorithmen benutzen die gesammelten Daten, um automatisch die relevanten Beziehungen in den Daten zu identifizieren. Darüber hinaus erstellen sie diese Modelle und Programmbefehle selbst. Das bedeutet, dass Benutzer keine vorhandenen Kenntnisse über den Prozess brauchen, um ein Model zu erstellen, und keine Programmierkenntnisse um Programmierbefehle zu schreiben. Dieses System ist automatisiert und flexibel. Das ermöglicht hochdimensionale Datensätze schnell zu behandeln, Umgang mit verschiedenen Arten von Daten (numerisch, nominal, Text und Bilder), Echtzeitverarbeitung, die Anzahl der benötigten Proben in bestimmten Fällen zu reduzieren und Zeit für die Modellierung zu sparen.

In dem vorgeschlagenen Schema werden die Daten erfasst und als Eingang für Algorithmen verwendet. Die Algorithmen erstellen das Modell und Programmierbefehle für Simulation. Das erzielte Modell wird dann für neue Daten verwendet.

Die Algorithmen können sowohl Regressionsmodelle, als auch Klassifikationsmodelle behandeln. Die Regressionsmodelle können zu den Berechnungen der Zerspanungswerte dienen. Die Klassifikationsmodelle kann man benutzen, um anhand der Zerspanungswerten das Werkzeugzustand nach der Bearbeitung zu prognostizieren.

In dieser Arbeit wurde neues Schema erfolgreich angewendet. Mit dem neuen Vorgang wurden die Regressionsmodelle für die Abhängigkeiten der Verschleißtiefe von der axialen Schnitttiefe und Oberflächenrauheit von der axialen Schnitttiefe und von der Verschleißtiefe. Es wurde auch Klassifikationsmodell für Bestimmung des Werkzeugzustands anhand der axialen Schnitttiefe erzeugt. Für die Erstellung diesen Abhängigkeiten wurden die effizientesten Algorithmen ausgewählt.

Die erzielten Modelle können für die Berechnungen implementiert werden, weil sie hohe Genauigkeit haben. Mit diesem Schema kann man weitere Modelle des Drehfräsens erzielen und die Untersuchung dieses Prozesses beschleunigen.

4.7.2 Schema zur Steuerung und Überwachung des Drehfräsens in der Produktion

Mit den entwickelten Algorithmen kann ein System für adaptive Steuerung der Zerspanungswerte und für die Prozessüberwachung in der Produktion beim Drehfräsen erstellt werden. Ähnliche Systeme wurden bereits für das Drehen [68], das Fräsen [69] und für additive Technologien [22] entwickelt.

Im Rahmen dieser Arbeit wurde das Konzept solchen Systems fürs Drehfräsens entwickelt. Das Schema dieses Systems ist in Abbildung 4-34 dargestellt.

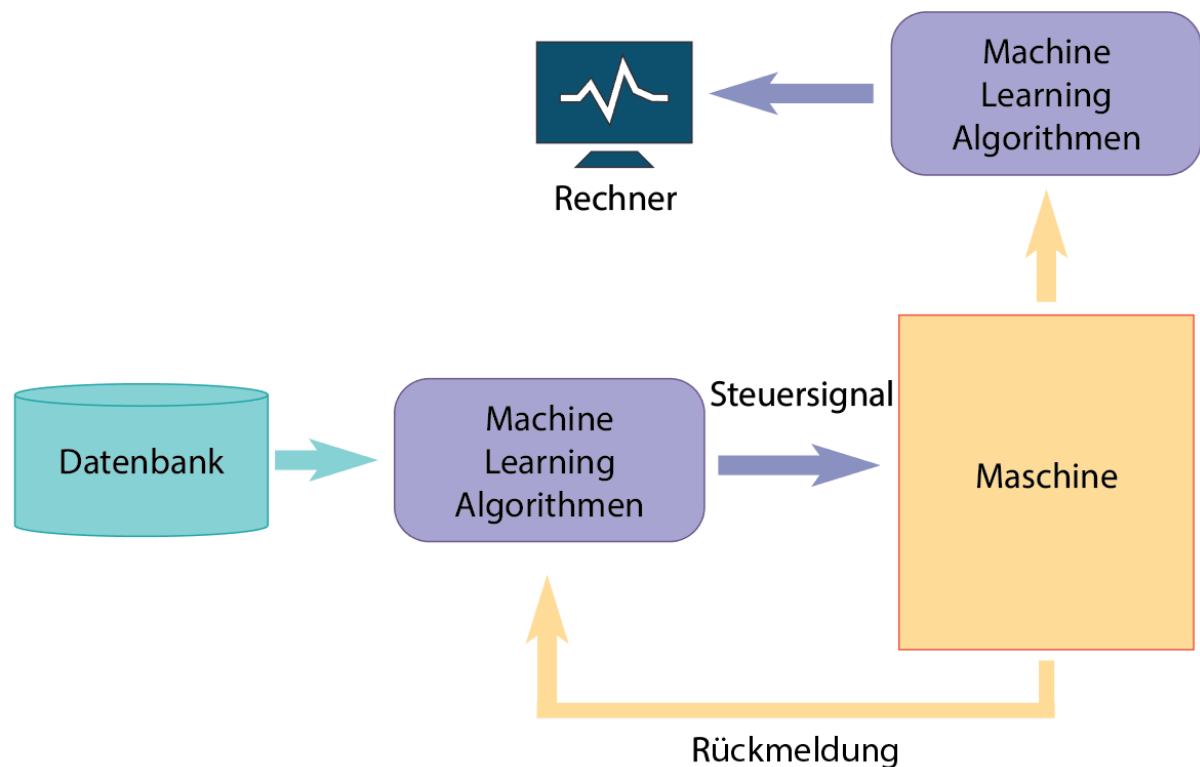


Abbildung 4-34: Konzipiertes Schema zur Steuerung und Überwachung des Drehfräsens

Dieses System besteht aus einer Datenbank, den Machine Learning Algorithmen, das Controller und den Sensoren. Am Anfang werden die Bearbeitungsdaten während des Fertigungsprozesses beim Drehfräsen erfasst. Dazu werden Teile mit unterschiedlichen Zerspanungswerten bearbeitet. Danach läuft die Qualitätskontrolle der erhaltenen Oberflächen und die Kontrolle des Werkzeugzustands nach der Bearbeitung ab. Diese Daten werden in einer Datenbank gespeichert, die fürs Lernen der ML-Algorithmen eingesetzt wird.

Dieses System verwendet eine Kombination von Regressionsalgorithmen, um ein modellbasiertes Steuersignal zu erzeugen. Algorithmen erhalten Eingaben von Sensoren, die sich in der Maschine befinden. Algorithmen können Ein-Parameter- oder Multi-Parameter sein. Die Hauptausschlusskriterien sind Einfachheit und Genauigkeit. Es ist besser, mehrere Ein-Parameter-Algorithmen als einen komplexen Algorithmus zu verwenden, wenn sie eine ausreichende Vorhersagegenauigkeit bieten. Somit funktioniert das adaptive Steuersystem.

Zusätzlich überwacht das System den Werkzeugzustand. Hierzu werden Klassifizierungsalgorithmen des maschinellen Lernens verwendet. Sie werden auch auf Datenbankbasis gelernt. Am Eingang erhalten sie Daten von den Sensoren und Zerspanungsdaten. Als nächstes zeigt das System Daten auf einem Computer an und warnt vor einem möglichen Werkzeugausfall. Dieses System funktioniert auch im Simulationsmodus. Es berechnet den gesamten Prozess mithilfe von Regressionsmodellen basierend auf ausgewählten Zerspanungsdaten und Arbeitsweg des Werkzeugs. Der

Klassifizierungsalgorithmus bestimmt dann, ob das Werkzeug dem gesamten Zyklus des Teils standhalten kann. Wenn das Werkzeug laut Berechnungen während der Verarbeitung ausgetauscht werden muss, warnt der Algorithmus den Maschinenbediener im Voraus.

Die entwickelten Algorithmen können bereits in einem solchen System implementiert werden, da sie im Validierungsprozess eine ausreichende Genauigkeit gezeigt haben. Um dieses System zu realisieren, muss die Datenbank mit anderen Parametern ergänzt werden: Bearbeitungslänge, Spindeldrehzahl, Vorschubgeschwindigkeit und Exzentrizität.

Dieses System macht das Drehfräsen kontrollierter. Dies wird eine breitere Verwendung dieser Bearbeitungsmethode in der Industrie ermöglichen.

In diesem Teil wurden das Optimierungsschema für die wissenschaftlichen Forschungen des Drehfräsens und das Schema zur Steuerung und Überwachung des Drehfräsens in der Produktion. Für diese Schemen wurden die Algorithmen entwickelt, validiert und ausgewählt. Sie haben gute Genauigkeit gezeigt und deshalb können die Algorithmen und schon erzielte Modelle in den konzipierten Schemen implementiert werden.

Zusammenfassung

In dieser Masterarbeit wurde das orthogonale Drehfräsen als Alternative zum herkömmlichen Drehprozess bei der Bearbeitung von Materialien für die Prothetik betrachtet. Derzeit ist dieses Verfahren trotz seiner Vorteile gegenüber dem Drehen (insbesondere bei der Bearbeitung harter Materialien) in der Produktion nicht weit verbreitet. Der Grund dafür ist nicht ausreichende Menge der Forschungen zu diesem Prozess und folglich schlechte Kontrollierbarkeit dieses Verfahrens. Deshalb besteht Bedarf an weiteren Untersuchungen des orthogonalen Drehfräsens.

Dazu wurden zwei Schemen unter Verwendung moderner Technologien der künstlichen Intelligenz vorgeschlagen. Sie dienen zur Optimierung der wissenschaftlichen Forschungen des Drehfräsens und beschleunigen die Masseneinführung dieses Verfahrens in Produktionssysteme.

Das erste Schema verwendet die ML-Algorithmen zur Automatisierung der Erstellung von Prozessmodellen. Diese Technik ermöglicht es nicht nur, Zeit bei der Entwicklung von Modellen zu sparen, sondern auch Geld, durch die Reduzierung der Anzahl von erforderlichen Experimenten. In dieser Arbeit wurden nach diesem Schema 4 Regressions- und eine Klassifizierungsmodelle fürs Drehfräsen anhand der experimentellen Daten erhalten. Zur Datenerfassung wurden Experimente durchgeführt, in deren Rahmen eine universelle CoCrMo-Legierung zur Herstellung von Prothesen aller Art bearbeitet wurde. Die resultierenden Modelle zeigten während der Validierung eine hohe Genauigkeit, was die Wirksamkeit dieser Technik bestätigt.

Die Algorithmen wurden als Programme mit einer benutzerfreundlichen Schnittstelle entwickelt. Somit können sie von anderen Forschern in der Fertigungstechnik verwendet werden. Dazu benötigen sie keine gründlichen Kenntnisse der Prinzipien von ML- Algorithmen, was die multidisziplinären Forschungen ermöglicht, ohne Spezialisten aus verschiedenen Bereichen einzubeziehen.

Das zweite Schema ist ein adaptives Steuersystem für die Zerspanungswerte und die Online-Überwachung des Werkzeugzustands. Entwickelte ML-Algorithmen gelten als Generator des Steuersignals und des Gehirns des Systems. Dieses System macht das Drehfräsen kontrollierter und reduziert Ausfallzeiten der Maschine. Der Vorteil dieses Systems ist die Fähigkeit zur Verbesserung und Umschulung (z. B. für eine neue Kombination von Werkzeugmaterial und Werkstückmaterial). Dazu muss die Systemdatenbank mit neuen Daten aufgefüllt werden. Somit ist das System sehr flexibel, was für die Agilität der modernen Produktion sehr wichtig ist.

Die in dieser Arbeit erhaltenen Modelle können aufgrund der hohen Genauigkeit, die während des Validierungsprozesses erhalten wird, in ein dieses System integriert werden. Für die vollständige Realisierung des vorgeschlagenen Systems ist es erforderlich, die Datenbank weiter aufzufüllen. Deshalb sind in der Zukunft Experimente geplant, um alle den Prozess beeinflussenden Parameter zu untersuchen.

Es wird empfohlen, die entwickelten Schemen und Programme fürs Drehfräsen im Rahmen des Konzepts „Industrie 4.0“ einzusetzen.

Literaturverzeichnis

- [1] BAHÇE, E., GÜLER, M. S. u. EMİR, E.: Investigation of Surface Quality of CoCrMo Alloy Used in the Tibial Component of the Knee Prosthesis According to the Methods of Turning and Turning-Grinding. *Materials Science* 26 (2019) 1, S. 41–48
- [2] M Pogacnik and J Kopac: Dynamic stabilization of the turn-milling process by parameter optimization. *Proc Instn Mech Engrs Vol. 214* (2000), S. 127–135
- [3] Choudhury, S.K. u. Mangrulkar, K.S.: Investigation of orthogonal turn-milling for the machining of rotationally symmetrical work pieces. *Journal of Materials Processing Technology* 99 (2000) 1-3, S. 120–128
- [4] Karagüzel, U., Uysal, E., Budak, E. u. Bakkal, M.: Analytical modeling of turn-milling process geometry, kinematics and mechanics. *International Journal of Machine Tools and Manufacture* 91 (2015), S. 24–33
- [5] The Royal Society: The AI revolution in scientific research. The Alan Turing Institute 2019
- [6] Figure 1a. Number of AI papers in Scopus by sub-category (1998-2017), 2019.000Z. https://www.researchgate.net/figure/a-Number-of-AI-papers-in-Scopus-by-sub-category-1998-2017_fig1_331471524, abgerufen am: 17.05.2020
- [7] AI investment & funding worldwide 2013-2018 | Statista, 2020.000Z. <https://www.statista.com/statistics/941137/ai-investment-and-funding-worldwide/>, abgerufen am: 17.05.2020
- [8] Moolayil, J.: An Introduction to Deep Learning and Keras. In: Moolayil, J. (Hrsg.): *Learn Keras for deep neural networks. A fast-track approach to modern deep learning with Python* / Jojo Moolayil. New York, NY: Apress 2019, S. 1–16
- [9] DATA STORY: Next investment opportunity? Artificial Intelligence market projected to grow 25-fold in eight years. *Moneycontrol* (09:20:12 AM IST 2017)
- [10] Russell, S. J. u. Norvig, P.: *Artificial intelligence. A modern approach*. Prentice Hall series in artificial intelligence. Englewood Cliffs N.J.: Prentice Hall 1995

- [11] artificial-intelligence noun - Definition, pictures, pronunciation and usage notes | Oxford Advanced Learner's Dictionary at OxfordLearnersDictionaries.com, 2020.000Z.
<https://www.oxfordlearnersdictionaries.com/definition/english/artificial-intelligence?q=artificial+intelligence>, abgerufen am: 17.05.2020
- [12] Theobald, O.: Machine Learning For Absolute Beginners: A Plain English Introduction (Second Edition). Scatterplot Press 2017
- [13] Burkov, A.: The hundred-page machine learning book. [S.l.]: Andriy Burkov 2019
- [14] Deng, L.: Deep Learning: Methods and Applications, Bd. 7. 2014
- [15] MassMutual Data Science, 2020.000Z.
https://datascience.massmutual.com/blog/smoker_propensity_model_using_deep_learning/, abgerufen am: 17.05.2020
- [16] VDMA Software and Digitalization: Quick Guide Machine Learning in Mechanical and Plant Engineering. Frankfurt am Main, Germany 2018
- [17] Wuest, T., Weimer, D., Irgens, C. u. Thoben, K.-D.: Machine learning in manufacturing: advantages, challenges, and applications. Production & Manufacturing Research 4 (2016) 1, S. 23–45
- [18] Wuest, T., Irgens, C. u. Thoben, K.-D.: An approach to monitoring quality in manufacturing using supervised machine learning on product state data. Journal of Intelligent Manufacturing 25 (2014) 5, S. 1167–1180
- [19] Lei Yu u. Huan Liu: Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. Proceedings, Twentieth International Conference on Machine Learning (2003), S. 856–863
- [20] Megahed, F. M. u. Camelio, J. A.: Real-time fault detection in manufacturing environments using face recognition techniques. Journal of Intelligent Manufacturing 23 (2012) 3, S. 393–408
- [21] Caggiano, A., Zhang, J., Alfieri, V., Caiazzo, F., Gao, R. u. Teti, R.: Machine learning-based image processing for on-line defect recognition in additive manufacturing. CIRP Annals 68 (2019) 1, S. 451–454
- [22] Delli, U. u. Chang, S.: Automated Process Monitoring in 3D Printing Using Supervised Machine Learning. Procedia Manufacturing 26 (2018), S. 865–870

- [23] Chowdhury, A., Kautz, E., Yener, B. u. Lewis, D.: Image driven machine learning methods for microstructure recognition. *Computational Materials Science* 123 (2016), S. 176–187
- [24] Gouarir, A., Martínez-Arellano, G., Terrazas, G., Benardos, P. u. Ratchev, S.: In-process Tool Wear Prediction System Based on Machine Learning Techniques and Force Analysis. *Procedia CIRP* 77 (2018), S. 501–504
- [25] Ko, T. J. u. Kim, H. S.: Autonomous cutting parameter regulation using adaptive modeling and genetic algorithms. *Precision Engineering* 22 (1998) 4, S. 243–251
- [26] KIRBY, E., CHEN, J. u. ZHANG, J.: Development of a fuzzy-nets-based in-process surface roughness adaptive control system in turning operations. *Expert Systems with Applications* 30 (2006) 4, S. 592–604
- [27] Edward Mehr REAL - TIME ADAPTIVE CONTROL OF ADDITIVE MANUFACTURING PROCESSES USING MACHINE LEARNING US PATENT 10 , 234 , 848 B2, Mar . 19 , 2019
- [28] Cuka, B. u. Kim, D.-W.: Fuzzy logic based tool condition monitoring for end-milling. *Robotics and Computer-Integrated Manufacturing* 47 (2017), S. 22–36
- [29] Cho, S., Asfour, S., Onar, A. u. Kaundinya, N.: Tool breakage detection using support vector machine learning in a milling process. *International Journal of Machine Tools and Manufacture* 45 (2005) 3, S. 241–249
- [30] Tsai, Y.-H., Chen, J. C. u. Lou, S.-J.: An in-process surface recognition system based on neural networks in end milling cutting operations. *International Journal of Machine Tools and Manufacture* 39 (1999) 4, S. 583–605
- [31] Huang, P. B., Ma, C.-C. u. Kuo, C.-H.: A PNN self-learning tool breakage detection system in end milling operations. *Applied Soft Computing* 37 (2015), S. 114–124
- [32] Krishnakumar, P., Rameshkumar, K. u. Ramachandran, K. I.: Tool Wear Condition Prediction Using Vibration Signals in High Speed Machining (HSM) of Titanium (Ti-6Al-4V) Alloy. *Procedia Computer Science* 50 (2015), S. 270–275
- [33] Apte, C., Weiss, S. u. Grout, G.: Predicting defects in disk drive manufacturing: A case study in high-dimensional classification. *Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications*. IEEE Comput. Soc. Press 1993, S. 212–218
- [34] Castejón, C., Lara, O. u. García-Prada, J. C.: Automated diagnosis of rolling bearings using MRA and neural networks. *Mechanical Systems and Signal Processing* 24 (2010) 1, S. 289–299

- [35] Marvuglia, A. u. Messineo, A.: Monitoring of wind farms' power curves using machine learning techniques. *Applied Energy* 98 (2012), S. 574–583
- [36] Sumesh, A., Rameshkumar, K., Mohandas, K. u. Babu, R. S.: Use of Machine Learning Algorithms for Weld Quality Monitoring using Acoustic Signature. *Procedia Computer Science* 50 (2015), S. 316–322
- [37] Content classification technology - Classifier - Antidot, 2020+00:00.
<https://www.antidot.net/classifier>, abgerufen am: 17.05.2020
- [38] Balic, J. u. Korosec, M.: Intelligent tool path generation for milling of free surfaces using neural networks. *International Journal of Machine Tools and Manufacture* 42 (2002) 10, S. 1171–1179
- [39] Famili, A.: Use of decision-tree induction for process optimization and knowledge refinement of an industrial process. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 8 (1994) 1, S. 63–75
- [40] Fichtner, D., Nestler, A., Kaldos, A. u. Rehm, C.: Neural network based application and data management for cutting values. In: Hayhurst, D. R., Hinduja, S., Atkinson, J., Burdekin, M., Hannam, R. G., Li, L. u. Labib, A. W. (Hrsg.): *Proceedings of the 33rd International MATADOR Conference. Formerly The International Machine Tool Design and Research Conference*. London: Springer London 2000, S. 113–118
- [41] Koltchinskii, V., Abdallah, C. T., Ariola, M. u. Dorato, P.: Statistical learning control of uncertain systems: theory and algorithms. *Applied Mathematics and Computation* 120 (2001) 1-3, S. 31–43
- [42] Margolis, D., Land, W. H., Gottlieb, R. u. Qiao, X.: A complex adaptive system using statistical learning theory as an inline preprocess for clinical survival analysis. *Procedia Computer Science* 6 (2011), S. 279–284
- [43] Comak, A. u. Altintas, Y.: Mechanics of turn-milling operations. *International Journal of Machine Tools and Manufacture* 121 (2017), S. 2–9
- [44] Choudhury, S. K. u. Bajpai, J. B.: Investigation in orthogonal turn-milling towards better surface finish. *Journal of Materials Processing Technology* 170 (2005) 3, S. 487–493
- [45] Karaguzel, U., Bakkal, M. u. Budak, E.: Process Modeling of Turn-Milling Using Analytical Approach. *Procedia CIRP* 4 (2012), S. 131–139

- [46] Schubert, A., Nestler, A. u. Funke, R.: Einfluss von Werkzeugschneidengeometrie und Prozessparametern auf die Oberflächenfeingestalt beim Drehfräsen von Aluminiummatrix-Verbundwerkstoffen. Influence of cutting edge geometry and cutting parameters on surface finish in turn milling of aluminium m. Materialwissenschaft und Werkstofftechnik 41 (2010) 9, S. 795–801
- [47] CTX beta 1250 TC - Turn & Mill by DMG MORI, 2020.000Z.
<https://en.dmgmori.com/products/machines/turning/turn-mill/ctx-tc/ctx-beta-1250-tc>,
 abgerufen am: 17.05.2020
- [48] Ergiloy 9.9135HL PROPRIETARY :: Total Materia, 2020.000Z.
<https://www.totalmateria.com/subgroup.aspx?LN=EN&id1=254887&db=N>, abgerufen
 am: 17.05.2020
- [49] Jugowiec, D., Łukaszczyk, A., Cieniek, Ł., Kot, M., Reczyńska, K., Cholewa-Kowalska, K., Pamuła, E. u. Moskalewicz, T.: Electrophoretic deposition and characterization of composite chitosan-based coatings incorporating bioglass and sol-gel glass particles on the Ti-13Nb-13Zr alloy. Surface and Coatings Technology 319 (2017), S. 33–46
- [50] Hoffmann, A. G.: General Limitations on Machine Learning. ECAI, S. 345–347
- [51] Aurélien Géron: Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media 2019
- [52] Jason Brownlee: Statistical methods for machine learning. Discover how to Transform Data into Knowledge with Python. eBook 2019
- [53] Shalev-Shwartz, S. u. Ben-David, S.: Understanding machine learning. From theory to algorithms. New York NY USA: Cambridge University Press 2014
- [54] Dinov, I. D.: Black Box Machine-Learning Methods: Neural Networks and Support Vector Machines. In: Dinov, I. D. (Hrsg.): Data science and predictive analytics [electronic resource]. Biomedical and health applications using R / Ivo D. Dinov. Cham.: Springer 2018, S. 383–422
- [55] RekhaMolala: Gradient descent algorithm and its three types | Clairvoyant Blog. Clairvoyant Blog (2019)
- [56] Chakrabarty, N.: Implementation of Uni-Variate Polynomial Regression in Python using Gradient Descent Optimization from scratch. Towards Data Science (2018)
- [57] Al-masri, A.: What Are Overfitting and Underfitting in Machine Learning? Towards Data Science (2019)

- [58] MODEL VALIDATION TECHNIQUES | Data Veda, 2020.000Z.
<https://www.dataveda.com/model-validation-techniques/>, abgerufen am: 17.05.2020
- [59] Awad, M. u. Khanna, R.: Efficient Learning Machines. Berkeley, CA: Apress 2015
- [60] Figure 5. Schematic of the one-dimensional support vector regression, 2019.000Z.
https://www.researchgate.net/figure/Schematic-of-the-one-dimensional-support-vector-regression-SVR-model-Only-the-points_fig5_320916953, abgerufen am: 17.05.2020
- [61] Opfer, G.: Numerische Mathematik für Anfänger. Wiesbaden: Vieweg+Teubner Verlag 2002
- [62] Swaminathan, S.: Logistic Regression — Detailed Overview - Towards Data Science. Towards Data Science (2018)
- [63] Figure 4. Schematic structure of a feed-forward artificial neural, 2019.000Z.
https://www.researchgate.net/figure/Schematic-structure-of-a-feed-forward-artificial-neural-network-model-and-the-equations_fig4_266838590, abgerufen am: 17.05.2020
- [64] Large Margin, 2020.000Z, abgerufen am: 17.05.2020
- [65] SVM for churn prediction - Machine Learning with Scala Quick Start Guide, 2020.000Z.
https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789345070/3/ch03lvl1sec30/svm-for-churn-prediction, abgerufen am: 17.05.2020
- [66] Role of Modeling and Simulation in Scientific Discovery, 2020.000Z.
<https://www.energy.gov/ne/articles/role-modeling-and-simulation-scientific-discovery>, abgerufen am: 17.05.2020
- [67] Traditional Statistics, 2020.000Z, abgerufen am: 17.05.2020
- [68] Solarte-Pardo, B., Hidalgo, D. u. Yeh, S.-S.: Cutting Insert and Parameter Optimization for Turning Based on Artificial Neural Networks and a Genetic Algorithm. Applied Sciences 9 (2019) 3, S. 479
- [69] Nestler, A. u. Schulz, G. (Hrsg.): Cutting values prediction with neural networks. 1999